Subject: Void pointer to NTL Array and VectorMap
Posted by manfhe on Thu, 29 Apr 2010 17:14:49 GMT
View Forum Message <> Reply to Message

Hello

I have a void pointer to an NTL Array and VectorMap, how do I receive the elements through the key?
The code looks like this:

```
typedef VectorMap<String, String> ntlMap;
typedef Array<String> ntlArray;

void* ptra;
void* ptrm;

ptra = new ntlArray;
ptrm = new ntlMap;

for(int i = 0; i < ((ntlMap*)ptrm)->GetCount(); i++)
 ((ntlArray*)ptra)->???? = ((ntlMap*)ptrm)->????;
```

And I would like to thank all contributors to this project. For beginners in C + + as I, U + + is a true teacher.

Grateful, Alan

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by dolik.rce on Thu, 29 Apr 2010 17:45:06 GMT
View Forum Message <> Reply to Message

Hi Allan,

You can always use operator[](). It looks a bit ugly, but it should work fine: for(int i = 0; i < ((ntlMap*)ptrm)->GetCount(); i++)
  ((ntlArray*)ptra)->operator[](i) = ((ntlMap*)ptrm)->operator[](i); Maybe there is a nicer way to do this, but in U++ there is so little cases where pointers are needed that I never really bothered to investigate any other possible solutions

Best regards,
Honza

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by manfhe on Thu, 29 Apr 2010 18:54:26 GMT
View Forum Message <> Reply to Message

Thanks Honza.

Work well.

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by hojtsy on Thu, 29 Apr 2010 20:16:40 GMT
View Forum Message <> Reply to Message

Hi, a shorter version is below. But there is usually a better alternative to using void* type in the first place.

```
for(int i = 0; i < ((ntlMap*)ptrm)->GetCount(); i++)
    (*(ntlArray*)ptra)[i] = (*(ntlMap*)ptrm)[i];
```

Regards,
Sandor

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by mirek on Thu, 29 Apr 2010 22:01:13 GMT
View Forum Message <> Reply to Message

[quote title=manfhe wrote on Thu, 29 April 2010 13:14]Hello

I have a void pointer to an NTL Array and VectorMap, how do I receive the elements through the key?
The code looks like this:

```
typedef VectorMap<String, String> ntlMap;
typedef Array<String> ntlArray;

void* ptra;
void* ptrm;

ptra = new ntlArray;
ptrm = new ntlMap;

for(int i = 0; i < ((ntlMap*)ptrm)->GetCount(); i++)
 ((ntlArray*)ptra)->???? = ((ntlMap*)ptrm)->????;
```

Or you can do this:

```
ntlMap& map = *(ntlMap *)ptrm;
ntlArray& array = *(ntlArray *)ptra;
for(int i = 0; i < map.GetCount(); i++)
    array[i] = map[i];
```

BTW, I only hope that you are doing something really special

Using 'new' should be reserved only for really desperate cases

Quote:
And I would like to thank all contributors to this project. For beginners in C + + as I, U + + is a true teacher.

Grateful, Alan

Only be aware that U++ will teach you something little bit different from "classic" approach...

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by manfhe on Thu, 29 Apr 2010 22:15:04 GMT
View Forum Message <> Reply to Message

One more doubt about void* type.

To destroy it is better: delete (ntlMap*)ptrm (This calls the destructor of the class?)
Or just: delete ptrm (So only destroy the pointer?)

Sorry, this is about a C + + and not U + +. I'm sorry, but I'm a beginner in C + +.

Thanks

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by mirek on Fri, 30 Apr 2010 08:46:03 GMT
View Forum Message <> Reply to Message

manfhe wrote on Thu, 29 April 2010 18:15One more doubt about void* type.

To destroy it is better: delete (ntlMap*)ptrm (This calls the destructor of the class?)

Yes. Once you cast it, compiler thinks it is pointer to ntlMap.

Once again, I hope that you are doing something really special. You should not use void pointers in 'normal' code. Actually, you should avoid pointers whenever possible.

Mirek

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by cbpporter on Fri, 30 Apr 2010 09:01:46 GMT
View Forum Message <> Reply to Message

Actually, I think it is an unwritten rule derived from years of practice both in C and C++ that you shouldn't use void pointers except for function parameters and some struct members, and even then only when there is a need to . It is generally better to use typed pointers.

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by manfhe on Fri, 30 Apr 2010 11:40:10 GMT
View Forum Message <> Reply to Message

Thanks.

I need use void* to manage different types at runtime, and my understanding in C++ void* is the unique solution.
I'm trying to make a crazy table, similar to a Lua table.

Subject: Re: Void pointer to NTL Array and VectorMap
Posted by dolik.rce on Fri, 30 Apr 2010 13:03:15 GMT
View Forum Message <> Reply to Message

I'm not sure how crazy is your crazy table, but in most cases such problems can be solved using Value type. See documentation (paragraph 6) for details.

Honza
