

Re: Is CArray best for this...

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2008-01/msg00837.html>

- *From:* "David Ching" <dc@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 17 Jan 2008 12:44:48 -0800
-

"Norbert Unterberg" <nunterberg@xxxxxxxxxxxxxxxxxxxxx> wrote in message news:e36cXXUWIHA.4712@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

I often have cases where the size of the array is not known when you start putting elements into it. Examples could be: parsing XML with a streaming parser and adding some found elements in an array, storing the lines of a text file, stroing the results of a database query (getting the size of a query result can be an expensive operation) etc.

Fair enough. Setting the initial size to be as large as what can be reasonably expected is what I do.

I disagree here. I do not see how in any way a `std::vector` can be "foreign" to windows. The only MFC like thing about CArray/CList I see is the naming convention. I can not remember any Windows/MFC API that expects or returns CArray or CList. On the contry, I think that CArray is more "foreign" as `std::vector` since `std::vector` is part of the C++ standard and CArray is not. What do you think is missing from `std::vector` to make it attractive to windows developers?

Nothing is missing, but what is there appeals to computer scientists and not necessarily to the domain-specific sensibilities of Windows app developers. Iterators have strange syntax, some STL construct (forget which one) needs to reference ".first" and ".second" is totally bizarre, etc. This is completely acceptable, and even desireable, for computer scientists, but not for app writers who are not interested in becoming experts in collections per se, only in that they allow them to write their apps.

See, use `std::vector` and you do not need any work around.

Re: Is CArray best for this...

The std containers and iterators look strange when you start using them, but their concept is much better thought out than the MFC collections.

I thought the same when I first saw the std library, but I changed my mind the better technology got me.

The thing of it is: these library writers have nothing else to do except make a good performing, usable class. Is it asking too much for them to make it sensible out of the box? Struggling to use it should not be the required price to make one "change his mind." .NET provides collections that make sense without necessarily becoming an expert in order to use them and persuade one of their inherent superiority. So it's possible, but for whatever reason, it is not a goal of STL, Boost, or from what I gather, any of the other C++ constructs being worked on right now. It can be perceived as arrogance on their part, and more and more that's what I'm seeing it as.

-- David

.