

Re: Can you fix this program? : C++ Dynamic Array Problems

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vc/2006-03/msg00149.html>

- *From:* "Carl Daniel [VC++ MVP]" <cpdaniel_remove_this_and_nospam@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 7 Mar 2006 07:15:57 -0800
-

Russell Mangel wrote:

I have written the following program using VS2005. The program is a Dynamic Array similar to System.Collections.ArrayList in .NET. The program works okay until I reach 65536, I can't seem to figure out why, as it seems my logic is working okay. I am a .NET programmer so I am not used to dealing with un-managed C++ code. Please criticize my code if you think it is poorly written.

The main criticism would be why write it at all? This code is spelled `std::vector<T>` in C++. There's simply no need to write code like this except for the learning exercise of it. A few other criticisms inline below.

This is the loop from main() that will blow up the program if you change 65536 to a larger value.

```
for(int i = 0; i < 65536; i++)
{
    folder.entryID = "Testing";
    fc->Add(folder);
}
```

Thanks
Russell Mangel
Las Vegas, NV

// Begin Code

```
#include "stdafx.h"
#include <iostream>

using namespace std;
```

Re: Can you fix this program? : C++ Dynamic Array Problems

```
struct Folder
{
char *entryID;
};
```

Should the above have a destructor? In your example, you're only assigning character literals to entryID, so the answer is NO, but in the real world, that might be a different story. If entryID needs to be delete[]'d, then you should define a destructor, assignment operator and copy constructor for this class.

```
class FoldersCollection
{
public:
FoldersCollection::FoldersCollection()
{
Count = 0;
Capacity = 0;
}
FoldersCollection::~~FoldersCollection()
{
delete []m_Folders;
}
int Count;
int Capacity;
```

You might want to wrap these fields in accessors. As is, a client can simply change your Count or Capacity, breaking your class from the outside.

```
void FoldersCollection::Add(Folder folder)
{
if(Capacity == 0)
{
m_Folders = new Folder[INITIAL_CAPACITY];
m_Folders[0].entryID = folder.entryID;
Count ++;
Capacity = INITIAL_CAPACITY;
}
else
{
if(Count < Capacity)
{
m_Folders[Count].entryID = folder.entryID;
Count ++;
}
else
{
```

Re: Can you fix this program? : C++ Dynamic Array Problems

```
printf("Resizing Array. Capacity: %d.\n", Capacity);
Resize();

m_Folders[Count].entryID = folder.entryID;
Count ++;
}
}
}
```

Rewrite the above:

```
void FoldersCollection::Add(Folder folder)
{
    EnsureCapacity(Count+1);
    m_Folders[Count++] = folder;
}
```

```
Folder* FoldersCollection::GetList()
{
    return m_Folders;
}
```

If you're going to simply expose the inner array, there's little point in even making a class since a client can simply run roughshod all over your array once you've returned a pointer. I'd reconsider this design.

```
private:
    Folder *m_Folders;
    Folder *m_Temp;
```

Eliminate the above member variable— m_Temp

```
static const int INITIAL_CAPACITY = 4;
void FoldersCollection::Resize()
{
    // Double Capacity
    int newCapacity = Capacity*=2;
    // Create new Array
    m_Temp = new Folder[newCapacity];
    // Copy elements
    for(int i = 0; i < Capacity; i++)
    {
        m_Temp[i].entryID = m_Folders[i].entryID;
    }
}
```

Re: Can you fix this program? : C++ Dynamic Array Problems

```
delete []m_Folders;
m_Folders = m_Temp;
Capacity = newCapacity;
}
```

Rewrite this:

```
void FoldersCollection::EnusreCapacity(int requiredCapacity)
{
if (requiredCapacity <= Capacity)
return;

int newCapacity = Capacity*2;
if (newCapacity < INITIAL_CAPACITY)
newCapacity = INITIAL_CAPACITY;
if (newCapacity < requiredCapacity)
newCapacity = requiredCapacity;

Folder* temp = new Folder[newCapacity];

if (Count > 0)
{
for (int i = 0; i < Count; ++i)
temp[i] = m_Folders[i];
delete[] m_Folders;
}

m_Folders = temp;
}

};
void main(void)
{
FoldersCollection *fc = new FoldersCollection;
Folder folder;

// Works okay up to 65536, change to larger value to crash program...
// What I am doing wrong...
for(int i = 0; i < 65536; i++)
{
folder.entryID = "Testing";
fc->Add(folder);
}
printf("Finished... Count: %d Capacity: %d \n", fc->Count,
fc->Capacity); delete fc;
}
```

Re: Can you fix this program? : C++ Dynamic Array Problems