Subject: Derivating from Vector<>
Posted by victorb on Sat, 03 Mar 2007 19:12:13 GMT
View Forum Message <> Reply to Message

I am trying to derivate a class from Vector<...> but I can't get it to work. Any help from the Upp community would be welcome.

Here is some sample code:

```
#include <Core/Core.h>

using namespace Upp;

class IntVector : public DeepCopyOption<IntVector, Vector<int> >
{
public:
 IntVector(){Cout() << "IntVector\n"; }

 virtual ~IntVector(){Cout() <<  "~IntVector\n";}

 IntVector(const IntVector &src, int) {
  ::new IntVector;
  Vector<int>(src, 0);
  name = src.name;
  Cout() << "DCC\n";
 }

 String name;

 String ToString(void) {
  String dump;

  dump << name << " ";

  if (IsPicked()) return dump << "Picked";

  for (int i = 0; i < GetCount(); i++) {
   dump << At(i) << " ";
  }
  return dump;
 }

};

CONSOLE_APP_MAIN
{
 Cout() << "iv\n";
 IntVector iv;
```

```
 iv.name = "IV";

 iv.Add(5);
 iv.Add(6);

 Cout() << "iv2\n";
 IntVector iv2(iv, 0);

 Cout() <<  iv.ToString() << "\n";
 Cout() <<  iv2.ToString() << "\n";

}
```

I expect iv2 to be equal to iv at the end. There is probably something wrong with the deep copy constructor but I really can't figure it out.

Victor
