Subject: Serialization technique
Posted by Dsonophorus on Thu, 21 Nov 2013 05:58:29 GMT
View Forum Message <> Reply to Message

This is an extremely tiny, general and powerful bit of code that I just wrote that supports the serialization of a majority of all types of data. It typically involves just inserting one line into a class to give it serialization support.

It could be generalized fairly easily to give all classes run time data member reflection and then have serialization as an implication of that data reflection... so still just one line added to a class. Generalized undo/redo capability is also nearly trivial once comprehensive serialization support is in place.

here is a working example of it:

```
class CClass;
class CTest { public:
mSERIAL{ Serial.IO(Val1).IO(Val2).IO(Val3).IO(Num).IO(hCl); return Serial; };
  CTest() { mSet(this,sizeof(this),0) };
  S32     Val1;
  S32     Val2;
  CVrt2   Val3;
  S32     Num[3];
  CClass* hCl;
};

class CClass { public:
  CClass() { mSet(this,sizeof(this),0) };
  mSERIAL{ Serial.IO(Val1).IO(Val2).IO(classes).IO(hCl); return Serial; };
  S32     Val1;
  CVrt2   Val2;
  CTest   classes[3];
  CClass* hCl;
};


int main()//(int argc, char** argv)
{
  CSerial SerIO;
  CClass C;
  CTest Tst, Magic;


  Tst.Val1 = 1; Tst.Val2 = 2; Tst.Val3.Set(30,31); Tst.Num[0]=0; Tst.Num[1]=10; Tst.Num[2]=20;
Tst.hCl = &C;
  C.Val1 = 101; C.Val2.Set(101,102); C.hCl = &C; C.classes[0].hCl = C.classes[1].hCl =
```

```
C.classes[2].hCl = 0;


  SerIO.Serialize( Tst, 0 );
  C.Val1= -101; Tst.Val2=-2;
  SerIO.Serialize( Tst, 1 );

  SerIO.Serialize( Tst, 0 ); //save it
  C.Val1= -101; Tst.Val2=-2;
  SerIO.Serialize( Tst, 1 ); //load it

  // here is some magic - I will clone the original serialization into a different root object and it will
  // automatically allocate, construct and link in new objects so it can reproduce the structure
  SerIO.Serialize( Tst, 0 ); //save it
  C.Val1= -101; Tst.Val2=-2;
  //all pointers within the entire network of serialized objects must be valid or NULL
  Magic.hCl = 0;
  SerIO.Serialize( Magic, 1 ); //load it
};
```

This is part of a project I am working on that is intended to give C++ full run time support including seamless integration of interpreted C++ with direct serialization of classes and code to storage or cross platform remote machines with JIT (just in time) compilation. I think it would be a strong addition to U++ to have a subset of those capabilities. Let me know if you think it looks useful.

Regards,
Mark Riphenburg

File Attachments

1) Serial.hpp, downloaded 454 times