
Subject: How to access base class function in this case?

Posted by [sinpeople](#) on Sat, 31 Oct 2020 13:05:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi folks,

I am studying the binary serialization in an inheritance hierarchy. In the code below, it seems the serialization methods (both `Serialize()` and `ToString()`) of the base class is never called. Is there any way to call it?

I want the serialization of an extended class also includes its immediate user-defined base class.

Thank you so much!

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

```
using namespace Upp;
```

```
struct Foo {
```

```
    int    q;
```

```
    String text;
```

```
    Vector<int> data;
```

```
void Serialize(Stream& s)
```

```
{
```

```
    s % q % text % data;
```

```
}
```

```
String ToString() const
```

```
{
```

```
    String s;
```

```
    s << "q = " << q << ", text = \"" << text << "\", data:";
```

```
    for(int i = 0; i < data.GetCount(); i++)
```

```
        s << ' ' << data[i];
```

```
    return s;
```

```
}
```

```
Foo() { q = 0; }
```

```
};
```

```
struct FooExt : Foo
```

```
{
```

```
    int nMessageID;
```

```
    VectorMap<String, int> scores;
```

```
void Serialize(Stream& s)
```

```
{
```

```
    s % nMessageID % scores;
```

```

}

String ToString() const
{
    String s;
    s << "nMessageID = " << nMessageID;
    for(int i = 0; i < scores.GetCount(); i++)
    {
        s << scores.GetKey(i) << ' ' << scores[i];
    }

    return s;
}
FooExt() { q = 0; }
};

CONSOLE_APP_MAIN
{
    FooExt foo;
    LoadFromFile(foo, ConfigFile("data.bin"));
    Cout() << "Loaded Foo: " << foo << '\n';
    foo.text = FormatIntRoman(foo.q);
    foo.q++;
    foo.data.Add(foo.q);
    foo.nMessageID = 1001;
    foo.scores.Add("Jack", 95);
    foo.scores.Add("Joe", 100);
    StoreToFile(foo, ConfigFile("data.bin"));
    Cout() << "Saved Foo: " << foo << '\n';
}

```

Best Regards
David

Subject: Re: How to access base class function in this case?
 Posted by [mirek](#) on Sat, 31 Oct 2020 13:39:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

```

struct FooExt : Foo
{

```

