
Subject: Re: Serialize of derived classes

Posted by [mdelfede](#) on Thu, 24 May 2012 11:43:47 GMT

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

Well... it should be enough to add a couple of templates :

```
template<class T> class WithPolyJson : public WithFactory<T>
template<class T> class WithPolySerial : public WithFactory<T>
```

And mimick the PolyXML series replacing Xmlize with Jsonize or Serialize.

ClassFactory template should stay unchanged, if you don't need extra features, but I guess I embedded enough of them

The best way to do it would be to add a PolySerialize package that uses my PolyXML, just for ClassFactory template.

I could also separate ClassFactory from PolyXML, if needed.

The most complicated part would be to mimick the PolyXMLUnknown class, which is used to stream in unknown objects keeping their xml code. Just seldom used, mostly if you've data from a newer version of your app that provides objects not foreseen in a previous version. It's able to stream in and out some pieces of unknown data without destroying them :

```
template<class T> class PolyXMLUnknown : public T
{
private:
    String tag;
    String rawXML;
public:
    PolyXMLUnknown(String const &_tag, String const &xml)
    {
        tag = _tag;
        rawXML = xml;
    }

    virtual String const &IsA(void) { return CLASSFACTORY_UNKNOWN; }
    String const &GetUnknownClassName(void) { return tag; }

    virtual void Xmlize(XmlIO &xml)
    {
        if(xml.IsStoring())
        {
            XmlNode node = ParseXML(rawXML);
            xml.Add();
            xml.Node() = node;
        }
    }
}
```

```
}  
};
```

You should find a way to do the same with Json and binary serialization, which I don't know if it's possible.

Max
