
Subject: Re: serialization

Posted by [mirek](#) on Sun, 02 Apr 2006 15:51:17 GMT

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

gprentice wrote on Sun, 02 April 2006 07:18

In the serialize reference example, this code appears to cleverly handle both the saving and restoring - is that correct?

```
void Serialize(Stream& s)
{
    s % q % text % data;
}
```

Yes.

Quote:

Suppose you have 100 items to save/restore - is there any reason not to chain all together in one long expression?

Usually not (see e.g. `ide/idewin.cpp` line 41)

Quote:

Comparing with .ini file, suppose some objects get removed - is there any alternative to having "dummy" objects of the correct type so that you can still read in data that was saved with now obsolete items?

Well, not really. Binary serialization has its limitations. In fact, I would not recommend it for "robust" data storage, e.g. for storing documents. We are using it for e.g. configuration files, passing data over network or backuping data for implementation reasons (e.g. Cancel in dialogs can be implemented using serialization - before stating dialog, its content is serialized into String, Cancel serializes it back).

Mirek
