Subject: Re: (Possible) Serialization issue
Posted by mirek on Sat, 01 Nov 2008 14:11:20 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sat, 01 November 2008 06:53luzr wrote on Mon, 04 August 2008 01:21Mindtraveller wrote on Sun, 03 August 2008 16:03Thanx. And what about renaming? I understand compitibility issues, but don`t you consider "no surprises" idea importance?Well, I do not see any surprise here. Well, I returned to old topic because could finally summarize what is really a surprise here. The main surprise is when Serialize() is called and you can`t determine if it is REALLY a loading process or a storing.

How so? IsLoading/IsStoring says exactly that. Is IsLoading is true, you should load from stream, store to it otherwise.

Quote:
Why is it a problem? Loading (and saving) process may be followed by lengthy GUI updates or some other initializations including file/database work.

Indeed (note however that is mostly the case for Loading).

Quote:
For example, on saving, my app generates a number of directories and files, then archives them with external utility. This was thought to be called rarely because it takes moderate amount of time and system resources (and these files have to be kept along a number of app runs). But each time I start application these files are re-created and re-archived because LoadFromFile() calls Serialize() with stream.IsStoring() == true. This situation will be complete surprise for those who use LoadFromFile() and didn`t read this topic.

OK, I understand the problem. However, Serialize is indended for storing into stream or retrieving. Nothing about archiving with external utility as part of Serialize body was ever considered.... This is the same as e.g. managing RS232 port in Paint routine...

Quote:
The solution here could be adding some additional function like stream.IsXXXXXXX() which will guide real internal serializarion direction to be used internally by U++, while IsStoring()/IsLoading() could return general "user-level" serialization direction (IsLoading() always == true on LoadFromFile()).

But there really is no such thing. Real internal serialization direction is IsStoring for the backup pass of LoadFromFile. And it really does STORE data into stream.

Frankly, if you want things like this, why do not you just call Serialize directly? Well, it is 2 lines instead of one, but really quite simple...

FileOut out(myfile);

x.Serialize(out);

Mirek