
Subject: Re: (Possible) Serialization issue
Posted by [mirek](#) on Wed, 30 Jul 2008 08:23:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mindtraveller wrote on Wed, 30 July 2008 01:14 Short description: while loading binary data from file, stream.IsStoring() returns true and stream.IsLoading() returns false.

```
Compatible testcase:
class TestClass
{
public:
    void Serialize (Stream &stream);

private:
    int b;
};

void TestClass::Serialize(Stream &stream)
{
    if (stream.IsStoring())
    {
        int a = 0; //BREAKPOINT STOP HERE!
    }
    if (!stream.IsLoading())
    {
        int a = 1; //BREAKPOINT STOP HERE!
    }

    stream % b;
}

CONSOLE_APP_MAIN
{
    TestClass testClass;
    LoadFromFile(testClass, ".config");
}
```

Recommended behaviour: in a number of situations it is vital to know if one loads or saves data. Correct IsStoring/IsLoading behaviour here is welcome.

It is a little bit more complicated.

This is "load failure" protection. LoadFromFile first **stores** data in temporary String buffer, then attempts to load them from the file. If load fails, it restores the backup copy.

```
bool Load(Callback1<Stream&> serialize, Stream& stream) {
    StringStream backup;
    backup.SetStoring(); // <<<< this is it :)
```

```
serialize(backup);
ASSERT(!backup.IsError());
stream.SetLoading();
stream.LoadThrowing();
try {
    CheckedSerialize(serialize, stream);
}
catch(LoadError) {
    backup.Seek(0);
    backup.SetLoading();
    serialize(backup);
    ASSERT(!backup.IsError());
    return false;
}
return true;
}
```

Note that if this is not what you want, do not use LoadFromFile

Mirek
