Subject: serialize and endians on non-x86 machines [BUG]
Posted by iplayfast on Fri, 24 Feb 2006 07:26:12 GMT
View Forum Message <> Reply to Message

I was tracing through the serialize example (wow this stuff is impressive).
One thing I noticed is that you are not taking endian into account. (Which is important for some processors.)

Something else which might be important to you when porting to different processors is some processors insist on even boundaries for integer values. That is structures like
pack(1)
struct S
{
char A;
int B;
};

Will cause an exception (and possible crash) when B is selected.

I've run into it when reading data from disk into a structure and then trying to access the data.

EDIT: new topic name by fudadmin

Subject: Re: serialize and endians
Posted by mirek on Fri, 24 Feb 2006 09:55:38 GMT
View Forum Message <> Reply to Message

iplayfast wrote on Fri, 24 February 2006 02:26I was tracing through the serialize example (wow this stuff is impressive).
One thing I noticed is that you are not taking endian into account. (Which is important for some processors.)

Something else which might be important to you when porting to different processors is some processors insist on even boundaries for integer values. That is structures like
pack(1)
struct S
{
char A;
int B;
};

Will cause an exception (and possible crash) when B is selected.

I've run into it when reading data from disk into a structure and then trying to access the data.

EDIT: new topic name by fudadmin

Yes, correct. We are well aware about the issue (Stream.cpp, line 558:

//#must be changed for nonIA32....
Stream& Stream::operator%(bool& d)
....

).

non-x86 archichitetures are one of themes for this year. However, I believe that it cannot be done correctly before we start testing on real HW... (which is to happen soon with ARM/WinCE).

BTW, "//#" comment is used thoughout the code to signal CPU dependent stuff.

Mirek
