Subject: Writing Float big-endian on Windows
Posted by RedDevil on Fri, 25 Feb 2011 01:04:54 GMT
View Forum Message <> Reply to Message

I am just starting to use U++ and find the experience very satisfying so far. I need to be able to read and write floats (SP) is big-endian format. I could not fine the equivalent to something like 'Put32be'. Any help would be appreciated - thanks.

Subject: Re: Writing Float big-endian on Windows
Posted by dolik.rce on Fri, 25 Feb 2011 06:19:55 GMT
View Forum Message <> Reply to Message

RedDevil wrote on Fri, 25 February 2011 02:04I am just starting to use U++ and find the experience very satisfying so far. I need to be able to read and write floats (SP) is big-endian format. I could not fine the equivalent to something like 'Put32be'. Any help would be appreciated - thanks.
Hi RedDevil

You can use the Put32be for this as well. All you have to do is to (carefully) cast the number into integer. Something like this should work:
stream.Put32be(*((int*)&f)); //where f is your float
Note that you must cast pointers, not the float directly, otherwise the value would be converted (and the bits would change). Similar trick is possible when reading as well, just the other way around.

Best regards,
Honza

Subject: Re: Writing Float big-endian on Windows
Posted by gprentice on Sat, 26 Feb 2011 05:28:42 GMT
View Forum Message <> Reply to Message

If you care about portability you shouldn't assume that int is 32 bits or that float is 32 bits. They don't have to be the same size.

Also, I don't know why stream provides big endian and little endian functions instead of choosing automatically but assuming that float is big endian isn't exactly portable either. If you use Get and Put I wonder why would you care about how it's stored?

Graeme

Subject: Re: Writing Float big-endian on Windows

Posted by dolik.rce on Sat, 26 Feb 2011 06:27:00 GMT
View Forum Message <> Reply to Message

Oups, gprentice is right about the int  In U++ you should use int32 or some similar type where the length is guaranteed. In standard c++ there is int32_t defined somewhere.

As for to reason for using certain endianness, I can see at least two  First, if you want to store data on one machine and read them on other, which has possible different architecture. Second, there is many standard file formats which have prescribed endianness (it is actually related to the first case), so that you have to obey it, even if you will always run it on the same machine, because otherwise other application wouldn't interpret the file correctly.

Honza

Subject: Re: Writing Float big-endian on Windows
Posted by RedDevil on Wed, 02 Mar 2011 23:57:42 GMT
View Forum Message <> Reply to Message

Thanks for the replies guys.  As Honza indicated I am writing to a format that is defined to be BigEndian, but doing it on a Intel PC.