Subject: streaming [FEATURE REQUEST]
Posted by amando1957 on Thu, 27 Nov 2008 17:38:34 GMT
View Forum Message <> Reply to Message

The String's are claimed to be fit for binary data, but do not provide the operator%.
The FileOut have got a Put(), but do not provide the operator<<.

Would not it be the best when String and FileOut both herit the << and %?
Alternative FileOut could get a constructor which allows to select binary or text, and also Little or Big Endian, which IMHO depends of the file format written.

Martin

Subject: Re: streaming [FEATURE REQUEST]
Posted by mirek on Fri, 28 Nov 2008 07:39:48 GMT
View Forum Message <> Reply to Message

amando1957 wrote on Thu, 27 November 2008 12:38The String's are claimed to be fit for binary data, but do not provide the operator%.

You can use StringStream.

The problem is that for String, it would to weird to support bidirectional operator%.

Using StringStream is almost as simple as using String proper.

(But you sent me thinking - maybe only the output is what we need there, right?).

Quote:
The FileOut have got a Put(), but do not provide the operator<<.

It does.

Quote:
Alternative FileOut could get a constructor which allows to select binary or text

And that would do what? Convert '\n' to "\r\n" on Windows?

BTW, such thing is extremely constly in terms of performance in certain scenarios.

Quote:
, and also Little or Big Endian, which IMHO depends of the file format written.

IMO it is more logical to use specific Putn[le|be]. LE/BE is usually not a variable for given file format.

Mirek

Subject: Re: streaming [FEATURE REQUEST]
Posted by amando1957 on Fri, 28 Nov 2008 20:00:10 GMT
View Forum Message <> Reply to Message

Hi Mirek

Quote:You can use StringStream.

Thank you.

Quote:maybe only the output is what we need there, right?
Yes, I consider string to be an outstream. String could also get a saveAs() function, then the FileOut would even get obsolete, for text.

Quote:Quote:
The FileOut have got a Put(), but do not provide the operator<<.
It does.

Excuse me, my mistake. I have overlooked.

Quote:Quote:
Alternative FileOut could get a constructor which allows to select binary or text
And that would do what?

Would save it as binary file or as text file ("wt" or "wb" with FILE pointer), no matter which operator was used for filling. At Windows there are both possible (unlike Linux I've heard).

Quote:Convert '\n' to "\r\n" on Windows?
at Windows I have always written \n to a text file and never had a problem, thus I don't see the need for.

Quote:IMO it is more logical to use specific Putn[le|be].

Do not know what you mean with Putn[le|be].

Quote:LE/BE is usually not a variable for given file format.

Hmmm, I am pretty sure it is. For a *.wav (LE) I need to write/read the low byte first, for a *.mid (BE) at last.
Now its not a problem to sort that myself and push the single bytes then. It would be just more convenient to put a dword LE or BE like:

StringStream binData(LE).
dword d = 0xAA2244FF;
binData % d;

But its your business of course how you make it ultimatly.

kind greets
Martin

Subject: Re: streaming [FEATURE REQUEST]
Posted by mirek on Fri, 28 Nov 2008 20:11:18 GMT
View Forum Message <> Reply to Message

Quote:
Would save it as binary file or as text file ("wt" or "wb" with FILE pointer), no matter which operator was used for filling. At Windows there are both possible (unlike Linux I've heard).

Yeah, but what is the difference?

There is no "wt" or "wb" in Linux, because in Linux there is no difference at all. In Windows, the difference is only in resolving that stupid '\r' issue. In U++, that is succesfully hidden by GetLine usually.

Quote:
Do not know what you mean with Putn[le|be].

Put32le, Get32be etc...

[quote]
Quote:LE/BE is usually not a variable for given file format.

Hmmm, I am pretty sure it is. For a *.wav (LE) I need to write/read the low byte first, for a *.mid (BE) at last.
[/quoute]

Variable = can be different for one format. It is either one or another (usually based on what system the format has emerged. Amiga / Mac orginated formats tend to be BE, PC formats LE).

There are of course exception to this rule, some formats can e.g. have flags that resolves this.

Quote:
StringStream binData(LE).
dword d = 0xAA2244FF;
binData % d;

But its your business of course how you make it ultimatly.

You should never read 'public' file formats using serialization, the way how things are serialized is "implementation detail".

Use those Getn[le|be] instead.

Mirek

Subject: Re: streaming [FEATURE REQUEST]
Posted by amando1957 on Sat, 29 Nov 2008 18:40:26 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Quote:Yeah, but what is the difference?
If it is text I can open it with the editor, otherwise not.

Thank you for the hints.
Martin

Subject: Re: streaming [FEATURE REQUEST]
Posted by mirek on Sat, 29 Nov 2008 18:55:20 GMT
View Forum Message <> Reply to Message

amando1957 wrote on Sat, 29 November 2008 13:40Hi Mirek,

Quote:Yeah, but what is the difference?
If it is text I can open it with the editor, otherwise not.

Actually, only as long as "editor" is notepad.exe. Other editors (including everything in U++) are able to deal with just '\n' instead of "\r\n" just fine.

BTW, U++ policy is to generate "\r\n" on output (but with the code) and ignore '\r' on input...

Mirek
