
Subject: Memory Mapped Files

Posted by [crydev](#) on Thu, 02 May 2013 07:38:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

I'm building a program where I need to write data from a list/vector to the memory with very high I/O performance, because it has to be very fast. In this case you have to be thinking of around ~1 GB of raw data.

I have been looking at memory mapped files. I think this is a very good solution for me. What I looked for first after I got it to know though, was looking for a solution embedded in U++. I couldn't find one. Is there a solution for using memory mapped files? Or do you guys/girls maybe have another solution then memory mapped files that I should have a look at?

Thanks in advance!

crydev

Subject: Re: Memory Mapped Files

Posted by [BioBytes](#) on Thu, 02 May 2013 18:30:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Crydev,

Did you have a look to MemStream or MemReadStream object ?

[http://www.ultimatepp.org/src\\$Core\\$Stream\\$en-us.html](http://www.ultimatepp.org/src$Core$Stream$en-us.html)

Regards

Biobytes

Subject: Re: Memory Mapped Files

Posted by [dolik.rce](#) on Thu, 02 May 2013 18:48:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi crydev,

crydev wrote on Thu, 02 May 2013 09:38 write data from a list/vector to the memory
Is that really what you wanted to write? If yes, then copying data from Vector to other location in memory can be as simple as calling `memcpy(dest, vector.Begin(), count*itemsize)`.

If you actually meant copying from/to file on disk, then Streams are probably the easiest way to go. AFAIK there is currently nothing using memory mapped files in U++. But to use it for storing, it should be just a simple call to `mmap` (on Linux) followed by `memcpy`.

Best regards,
Honza

Subject: Re: Memory Mapped Files
Posted by [mirek](#) on Sat, 11 May 2013 08:02:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

U++ actually has file mapping encapsulated to hide platform differences, see FileMapping.
