Subject: NEW: BufferStream
Posted by kohait00 on Wed, 11 Aug 2010 09:25:17 GMT
View Forum Message <> Reply to Message

hi guys,

i was in need of a BufferStream, a stream that can store data inside a Vector<byte>, which i find beeing more convenient than the StringBuffer, which is quite hard to understand (having both a String data and a StringBuffer etc..where data is copied around i think..which is not what i needed). BTW: why was that nessessary, why couldnt it have been only StringBuffer or only String data?

so i tried to use my brandnew Stream knowledge to make a Vector<byte> based buffer, maybe one could take a look on it and spot some conceptional mistakes, if any..(ofcorse there are:)

it should support storing data and GetResult a Vector<byte> back, picking internal one, which then is resetet to be used again. it reserves some more space in advance..which is not equal to currently stored stuff there..so this is definitely fix issue yet.

maybe a i only dont understand the StringBuffer well, maybe it could do it for me as well. but for for now it is my inly option. if somebody can explain the internal concept of StringBuffer, i'd appreciate it.. so here comes the class, attached a Test app.

```
class BufferStream : public MemStream {
protected:
 virtual  void  _Put(int w) { byte h = w; _Put(&h, 1); }
 virtual  void  _Put(const void *data, dword size)
 {
  if(size > (dword)(uintptr_t)(wrlim - ptr)) {
   Reserve(size + 128);
  }
  memcpy(ptr, data, size);
  ptr += size;
 }

public:
 virtual  void  SetSize(int64 asize)
 {
  dword size = (dword)asize;
  dword p = (dword)(uintptr_t)(ptr - buffer);
  data.SetCount(size);
  Open(data);
  SetStoring();
  Seek(min(p, size));
 }

protected:
 Vector<byte> data;

public:
 void      Open(Vector<byte> & d)
 {
  if(&data != &d) data = d; //pick
  MemStream::Create((byte*)data, data.GetCount());
 }

 void       Create()
 {
  data.Clear();
  Open(data);
  SetStoring();
 }

 void       Reserve(int n)
 {
  SetSize((int)GetSize() + n);
 }

 Vector<byte> GetResult()
 {
  Vector<byte> d = data; //pick
  Create();
  return d;
 }
 operator    Vector<byte>()              { return GetResult(); }

 BufferStream()                          { Create(); }
 BufferStream(Vector<byte>& d)           { Open(d); }
};

typedef BufferStream VectorStream;
```

EDIT: luckily noone has downloaded this one so far..
so i exchange it with an update..GetResult now trims data to only used..

File Attachments

1) BufferStream.rar, downloaded 183 times

Subject: Re: NEW: BufferStream
Posted by mirek on Fri, 13 Aug 2010 08:57:11 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Wed, 11 August 2010 05:25hi guys,

i was in need of a BufferStream, a stream that can store data inside a Vector<byte>, which i find beeing more convenient than the StringBuffer, which is quite hard to understand (having both a String data and a StringBuffer etc..where data is copied around i think..which is not what i needed). BTW: why was that nessessary, why couldnt it have been only StringBuffer or only String data?

Because sometimes the code uses Stream interface.

E.g. you have Serialize and you want to store the output to String (or read the input).

Subject: Re: NEW: BufferStream
Posted by kohait00 on Fri, 13 Aug 2010 09:02:33 GMT
View Forum Message <> Reply to Message

didn't quite get it yet, sorry, which code? Stream interface is used always, or not?

Subject: Re: NEW: BufferStream
Posted by mirek on Fri, 13 Aug 2010 19:48:54 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Fri, 13 August 2010 05:02didn't quite get it yet, sorry, which code? Stream interface is used always, or not?

Now I see, you wanted to add Stream interface to String?

Well, that is not a good idea for many reasons...

Subject: Re: NEW: BufferStream
Posted by kohait00 on Sun, 15 Aug 2010 06:14:59 GMT
View Forum Message <> Reply to Message

why? selfgrowing buffers? forgetting to flush? reallocation?