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 214 times