

Exiv2 - Bug #469

Memlo behaviour differs from FILE* and Filelo

30 May 2006 10:11 - Andreas Huggel

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Andreas Huggel	% Done:	0%
Category:	miscellaneous	Estimated time:	0.00 hour
Target version:	0.10		

Description

When using readMetadata() on a TIFF that has been opened using Exiv2::ImageFactory::open(ptr,size) I always get an exception because EOF is reached. I think this is expected however because the relevant code just read in a number of bytes equal to the amount of the stream's size (tiffimage.cpp:133).
The exception only occurs when using Memlo, Filelo doesn't return true for eof() after reading in size() bytes.

(Reported by Dimitri)

Thanks, this looks like a bug in Memlo. A quick read test with a buffer and file of the same size gives this:

```
$ ./try
Testing 10 bytes
Memlo : .....
Filelo: .....
FILE  : .....

// -----
#include "basicio.hpp"
#include <iostream>
#include <cstdio>

using namespace Exiv2;

int main() {
DataBuf buf(10);
FILE* f = fopen("ttt", "wb");
std::cout << "Testing "
<< fwrite(buf.pData_, 1, buf.size_, f)
<< " bytes\n";
fclose(f);

Memlo mlo(buf.pData_, buf.size_);
std::cout << "Memlo : ";
while(!mlo.eof()) {
mlo.getb();
std::cout << ".";
}
std::cout << "\n";

Filelo flo("ttt");
flo.open();
std::cout << "Filelo: ";
while(!flo.eof()) {
flo.getb();
std::cout << ".";
}
}
```

```
std::cout << "\n";

f = fopen("ttt", "rb");
std::cout << "FILE : ";
while(!feof(f)) {
fread(buf.pData_, 1, 1, f);
std::cout << ".";
}
std::cout << "\n";

return 0;
}
```

History

#1 - 02 Jun 2006 06:20 - Andreas Huggel

[r810](#)