DCMTK - Bug #857

Implementation and documentation of calcElementLength() are inconsistent

2018-11-26 11:11 - Jörg Riesmeier

Status:	Closed	**Start date:**	2018-11-26
Priority:	Normal	**Due date:**	
Assignee:	Jan Schlamelcher	**% Done:**	100%
Category:	Library	**Estimated time:**	0:00 hour
Target version:	3.6.6		
Module:	dcmdata	**Compiler:**	
Operating System:			

Description

Currently, both the implementation and the documentation of method calcElementLength() are inconsistent: Some classes use DCM_UndefinedLength and set the internal "erroFlag" in order to indicate an integer overflow (32-bit length field), others use OFnumeric_limits<Uint32>::max(), which happens to be the same value as DCM_UndefinedLength, but do not set the internal flag. The latter (OFnumeric_limits<>) is wrong (since it is an odd value), especially since the errorFlag is not set. Also the documentation of some derived classes is incorrect, claiming a different implementation (see e.g. DcmElement, DcmDataset, DcmPixelItem, DcmPixelSequence).

Using @copydoc, which currently always points to the base class DcmObject, makes things even worse...

Related issues:

Related to DCMTK - Feature #806: Maximum value returned by calcElementLength(...	**New**	**2017-12-13**
Precedes DCMTK - Bug #951: Refactor calcElementLength()	**New**	

History

#1 - 2020-05-25 13:29 - Michael Onken

- Target version deleted (3.6.6)

#2 - 2020-05-25 13:36 - Michael Onken

- Target version set to 3.6.6

#3 - 2020-10-01 13:19 - Jan Schlamelcher

- Status changed from New to Resolved

- % Done changed from 0 to 40

Well, I have to agree the current implementation of calcElementLength() in the various classes of the DcmObject hierarchy is an absolute clusterfuck. I still believe the only truly reasonable implementation would be to return the result in a more appropriate data type, e.g. size_t, so we simply would not have to cut it off artificially.

Regarding the points criticized specifically: sorry, but this is all over the place. It took me quit some time to understand what problems you actually referred to. What I found out is, that there are three separate issues with the current implementation:

1. The documentation is inconsistent, since not all header files were updated when the code was changed last time. I will address this by rewriting the documentation of DcmObject::calcElementLength() to describe it correctly for all derived classes and then use @copydoc everywhere else to make sure that never happens again.

2. The calcElementLength() function actually never sets the internal error flag (except for the implementation of DcmPixelData, that seems to do its own thing entirely). In some cases, however, calcElementLength() is based on an implementation of getLength() that does that in case an item's or a sequence's length cannot be represented as a 32bit number such that it must be encoded as undefined length, but encoding as undefined length was disabled, so it would not be possible to encode the value when writing a DICOM dataset. This is an entirely different case.

3. The implementations that return OFnumeric_limits<Uint32>::max() do so to indicated that the value can actually be represented, just not using the API we currently provide. This implementation is not ideal, but, since it coincides with the value of DCM_UndefinedLength, no problems can be expected but we still mark the places in the code where we will need something better in the future. Returning undefined length would just be wrong, since the length is defined, it just cannot be represented correctly due to the API limitation.

Since the title of this issue focuses on the documentation / implementation inconsistency, I will fix the documentation as described above and open a new ticket about refactoring the API to something sane.

#4 - 2020-10-01 13:51 - Jan Schlamelcher

- Status changed from Resolved to Closed

Closed by commit #fc8824e5a7f96f9884b.

#5 - 2020-10-01 13:52 - Jan Schlamelcher

- % Done changed from 40 to 100

#6 - 2020-10-01 14:14 - Jörg Riesmeier

- Precedes Bug #951: Refactor calcElementLength() added

#7 - 2020-10-01 14:15 - Jörg Riesmeier

- Related to Feature #806: Maximum value returned by calcElementLength() is limited to 2^32-1 added