

DCMTK - Bug #845

compression followed by decompression can lead to inconsistent datasets

2018-08-16 15:30 - Marco Eichelberg

Status:	Closed	Start date:	2018-08-16
Priority:	Normal	Due date:	
Assignee:	Marco Eichelberg	% Done:	100%
Category:	Library	Estimated time:	3:00 hours
Target version:	3.6.6	Compiler:	
Module:	dcmdata, dcmjpeg, dcmjpls, dcmjp2k		
Operating System:			
Description			
Class DcmPixelData in dcmdata can store multiple "versions" of the pixel data element, one for each combination of transfer syntax and representation parameter.			
Unfortunately in some cases, a modification of the DICOM header and a modification of the raw pixel data is required when compressing an image (e.g. extraction of embedded overlays prior to lossy compression). When such an image is first compressed and then decompressed by the same application (using DcmDataset::chooseRepresentation()), the decoder is not called at all, the header remains unmodified and the original pixel data as it was before compression is re-activated. This may lead to inconsistent objects when changes were applied to the PixelData content prior to compression, because these changes will be lost.			
We need a new API call in class DcmPixelData that can be called by the various compression encoders in order to "invalidate" all representations of the pixel data other than the current (compressed) one. This will force dcmdata to use the decoder when access to uncompressed pixel data is requested, thus leading to a consistent dataset.			

History

#1 - 2018-11-15 19:27 - Jörg Riesmeier

- Target version changed from 3.6.4 to 3.6.6

#2 - 2020-05-25 13:29 - Michael Onken

- Target version deleted (3.6.6)

#3 - 2020-05-25 13:36 - Michael Onken

- Target version set to 3.6.6

#4 - 2020-05-27 16:26 - Marco Eichelberg

- Category set to Library
- Status changed from New to Closed
- Assignee set to Marco Eichelberg
- % Done changed from 0 to 100
- Estimated time set to 3:00 h

Closed by commit #0e8fae86c (public modules) and #84ccf502f (private modules).