

DCMTK - Feature #280

Viele Tools verwenden loadAllDataIntoMemory() beim Laden von Dateien

2008-07-21 00:00 - Marco Eichelberg

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0:00 hour
Target version:		Compiler:	
Module:	alle		
Operating System:			
Description			
Viele Tools rufen beim Laden von Dateien loadAllDataIntoMemory() auf, um sicherzustellen, dass ein Überschreiben der Datei korrekt funktioniert (wenn Eingabedateiname = Ausgabedateiname). Dadurch können alle diese Tools mit sehr großen DICOM-Dateien (mehrere GByte) aber nicht korrekt umgehen, selbst wenn die zugrundeliegenden Funktionen das eigentlich könnten. Daher muss dies soweit wie möglich reduziert werden. Aktuell betroffen sind folgende Tools: dcmdata/apps/mdfdsman.cc dcmdata/apps/dcmdrle.cc dcmdata/apps/dcmdump.cc dcmdata/apps/dcm2xml.cc dcmdata/apps/dcmconv.cc dcmdata/apps/dcmcrle.cc dcmjp2k/apps/dcmcj2k.cc dcmjp2k/apps/dcmdjp2k.cc dcmjpeg/apps/dcmcjpeg.cc dcmjpeg/apps/dcmdjpeg.cc dcmjpls/apps/dcmcjpls.cc dcmjpls/apps/dcmdjpls.cc dcmmisc/apps/dcmfixmg.cc dcmmisc/apps/dcmanon.cc dcmmisc/apps/dcmhmod.cc dcmmisc/apps/dcmhmod.cc dcmmisc/apps/dcmgramp.cc dcmmisc/apps/dcmclip.cc dcmsign/apps/dcmsign.cc dcmcheck/apps/dcmcheck.cc dcmimage/apps/dcmquant.cc dcmimage/apps/dcmscale.cc dcmpstat/apps/dcmpschk.cc Wenn man den Aufruf von loadAllDataIntoMemory() entfernt, muss man überlegen, was passiert, wenn tatsächlich jemand versucht, eine Datei auf sich selbst zu schreiben. Drei Möglichkeiten sehe ich: * Fehlermeldung und Ende\\ (schlecht: ändert die Semantik der Tools) * loadAllDataIntoMemory() nur dann aufrufen, wenn Eingabename=Ausgabename\\ (schlecht: die Funktionsfähigkeit der Tools hängt bei großen Dateien vom gewählten Ausgabedateinamen ab) * schreiben in Temporärdatei, nach erfolgreichem Schreiben Löschen der alten Datei und umbenennen der neuen.\\ (einziger Nachteil: kurzfristig wird mehr Speicherplatz belegt, ansonsten aber nach meinem "Gefühl" die sinnvollste Lösung, für die man ggf. eine Support-Routine in OFStandard o.ä. schreiben könnte) Falls eine andere Lösung als die dritte gewählt werden sollte, brauchen wir eine Routine, die entscheidet, ob Ausgabedatei = Eingabedatei ist, was bei symbolischen Links usw. nicht trivial ist. Hierzu einige Links: * http://idlastro.gsfc.nasa.gov/idl_html_help/FILE_SAME.html * http://lists.apple.com/archives/Darwin-dev/2005/Dec/msg00085.html * man 2 stat (liefert st_ino and st_dev); man 3 realpath * http://blogs.msdn.com/michkap/archive/2005/10/17/481600.aspx * http://msdn2.microsoft.com/en-us/library/aa364963.aspx (GetFullPathName()) * http://msdn2.microsoft.com/en-us/library/aa365247.aspx [W32 file name rules] === Kommentar JR ===			

Die Routine `DcmOtherByteOtherWord::print()` könnte auch dahingehend geändert werden, daß wenn das `DCMTypes::PF_shortenLongTagValues` Flag gesetzt ist, nicht der gesamte Elementwert "geladen" wird, sondern nur die benötigten ersten Bytes (`DCM_OptPrintLineLength`).