

DCMTK - Bug #390

Memory leaks in storescu, findscu und Co.

2011-08-25 00:00 - Jörg Riesmeier

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	50%
Category:		Estimated time:	0:00 hour
Target version:		Compiler:	
Module:	dcmnet (und weitere?)		
Operating System:			
Description			
found with valgrind on caesar It seems that 'ASC_destroyAssociation()' is not called in certain places in order to free the memory associated with the 'assoc' structure. This manifests itself in erroneous behaviour (e.g. unsuccessful negotiation of Presentation Contexts, no connection with the Communication Partner). Associated Patch: "Added new class template "OFResourceHandle" for RAII-style (leak free) programming and used it to fix some memory leaks in storescu." T_ASC_Association_Handle is introduced in this template, which includes ASC_destroyAssociation's functionality. If you replace all instances of T_ASC_Association with T_ASC_Association_Handle, ASC_destroyAssociation be called for each object. In some parts of the code, one must be careful as get() will be called on this handle to get the pointer. In most cases the overloaded operators in OFResourceHandle should ensure that the handle will contain the pointer as before and not require that the code be adapted. TODO: - So far this has only been fixed in storescu, needs to be done for the others - Test if all leaks have been fixed - Adjust the scope of T_ASC_Association_Handle objects so that resources are released as early as possible and not at the end of main.			
Related issues:			
Related to DCMTK - Bug #832: another echo SCU false positive		Closed	2018-05-03 2018-05-04
Related to DCMTK - Bug #831: echo SCU false positive		Closed	2018-05-03 2018-05-03

History

#1 - 2012-08-01 18:08 - Jan Schlamelcher

- File ofhand.patch added
- % Done changed from 0 to 50

Beiliegender Patch: "Added new class template "OFResourceHandle" for RAII-style (leak free) programming and used it to fix some memory leaks in storescu." Mit Hilfe dieses templates wurde das neue struct T_ASC_Association_Handle erstellt, was im destructor ASC_destroyAssociation ausführt. Somit wird diese funktion garantiert für jedes T_ASC_Association "Objekt" aufgerufen, sofern man alle Vorkommnisse von T_ASC_Association* variablen durch T_ASC_Association_Handle variablen ersetzt. An manchen stellen muss allerdings der code in soweit angepasst werden, dass get() von diesem handle aufgerufen wird um den normalen pointer zu bekommen. An den meisten stellen sorgen die in OFResourceHandle überladenen operatoren jedoch dafür, dass sich das handle genau so wie der pointer verhält und kein code angepasst werden muss.

- TODO:
- Bisher wurde dieser fix nur in storescu angewand, muss also noch bei den Anderen gemacht werden.
 - Testen ob alle leaks behoben sind.
 - Eventuell den scope der T_ASC_Association_Handle Objekte anpassen, damit die resourcen möglichst früh freigegeben werden und nicht erst am Ende der main.

#2 - 2013-02-13 17:20 - Jörg Riesmeier

- Target version set to 3.6.2

#3 - 2013-04-28 23:47 - Andrew Chiw

- Subject changed from Speicherlecks in storescu, findscu und Co. to Memory leaks in storescu, findscu und Co.

#4 - 2015-12-01 17:36 - Jan Schlamelcher

- Assignee set to Anonymous

#5 - 2017-03-24 12:42 - Marco Eichelberg

- Priority changed from High to Normal

- Target version changed from 3.6.2 to 3.6.3

#6 - 2017-12-07 11:58 - Marco Eichelberg

- Target version changed from 3.6.3 to 3.6.6

#7 - 2018-05-03 17:06 - Pedro Fernando Arizpe Gómez

- Related to Bug #832: another echo SCU false positive added

#8 - 2018-05-03 17:07 - Pedro Fernando Arizpe Gómez

- Related to Bug #831: echo SCU false positive added

#9 - 2018-10-22 11:46 - Marco Eichelberg

- Assignee deleted (Anonymous)

#10 - 2020-05-25 13:28 - Michael Onken

- Target version deleted (3.6.6)

Files

ofhand.patch	8.26 KB	2012-08-01	Jan Schlamelcher
--------------	---------	------------	------------------