DCMTK - Bug #443

storescp's option --inetd and --exec-on-eostudy are probably mutually exclusive

2012-09-14 09:01 - Jörg Riesmeier

Status:	Closed		**Start date:**	2012-09-14
Priority:	Normal		**Due date:**	
Assignee:	Uli Schlachter		**% Done:**	0%
Category:	Application		**Estimated time:**	0:00 hour
Target version:	3.6.2			
Module:	dcmnet		**Compiler:**	
Operating System:				

Description
Check whether this combination of options should be forbidden by storescp (pretty much like --fork and --exec-on-eostudy on Windows systems). See the following forum posting for details: http://forum.dcmtk.org/viewtopic.php?f=1&t=3534&p=14204#p14204

History

#1 - 2012-09-17 09:09 - Jörg Riesmeier

Also see this posting: http://forum.dcmtk.org/viewtopic.php?f=1&t=3545

#2 - 2012-09-21 17:28 - Uli Schlachter

- Status changed from New to Resolved

Fixed with commit d6b29fe15455da02927c. Solution was just to make --inetd and --exec-on-eostudy conflicting. The problem here is that the end of study detection works across different associations. Thus if you use storescu twice to transfer two images that are in the same study with two separate associations, they will only be counted as a single end of study. For this reason, the code doesn't trigger an end of study when an association is released. This problem also affects --fork on unix (tested with "storescp --exec-on-eostudy 'date;echo eostudy #p' --exec-on-reception 'date;echo reception #p' 1234 --output-directory /tmp/ --sort-conc-studies dcm-foo --fork --eostudy-timeout 1", no study is ever ended).

It might make sense to fix this differently in the future:
When we are exiting and a study end is still pending (lastStudySubdirectoryPathAndName is not empty), just execute that study end now. This means that with --fork and --inetd studies across different associations can't be detected, but at least it doesn't miss any studies and --exec-on-eostudy can be used.
(Patches welcome :P)

#3 - 2012-11-27 10:43 - Uli Schlachter

- Status changed from Resolved to Closed