U++ - Bug #803
Bad Clear() call to a const char *

07/05/2014 04:13 PM - Iñaki Zabala

Status:	Rejected		Start date:	07/05/2014
Priority:	Normal		Due date:	
Assignee:	Iñaki Zabala		% Done:	0%
Category:	Core		Estimated time:	0.00 hour
Target version:			Spent time:	0.00 hour

Description

In Socket.cpp, line 893, errordesc is a const char *, however errordesc.Clear() is called.

void TcpSocket::SetSockError(const char *context, int code, const char *errdesc){
    errorcode = code;
    errordesc.Clear();

History

#1 - 07/06/2014 04:54 PM - Miroslav Fidler

- Status changed from New to In Progress

- Assignee changed from Miroslav Fidler to Iñaki Zabala

I see "errdesc" being const char *. "errordesc" is String member of TcpSocket.

Have I got it wrong?

#2 - 07/07/2014 09:04 AM - Iñaki Zabala

- Assignee changed from Iñaki Zabala to Miroslav Fidler

Hello Mirek

However errordesc is also an argument in SetSockError. Please check the argument list:

void TcpSocket::SetSockError(const char **context, int code, *const char *errdesc**) {
    errorcode = code;
    errordesc.Clear();

#3 - 07/07/2014 09:15 AM - Miroslav Fidler

- Assignee changed from Miroslav Fidler to Iñaki Zabala

Please check carefully what is there. It is "errdesc", not "errordesc". Besides, it would compile, would it?

#4 - 07/08/2014 08:52 AM - Iñaki Zabala

- Status changed from In Progress to Rejected

Oups!

The problem came in errordesc.Clear();, but the reason was that my program indirectly deleted a TcpSocket used by an opened thread. Accidentally, the IsAbort() fired and the deleted TcpSocket went to SetSockError() where it finally exploded in the Clear() (as TcpSocket errordesc was previously deleted)...

Now I have changed the main class destructor so it cares of Abort() all TcpSocket, checks that all processes are ended and then, deletes itself.

Thank you.