Subject: Clean Exit
Posted by mirek on Mon, 21 May 2012 12:48:55 GMT
View Forum Message <> Reply to Message

New function

void Exit(int code = 1);

(Core/App.h)

is equivalent of C exit function, but implemented using exception (and catch in _APP_MAIN). This has an advantage of destructors being correctly called -> no leaks.

In related news, _APP_MAIN now catches Exc derivated exceptions and shows them using Panic.

For Rainbow implementation, GUI_APP_MAIN should now use AppExecute__(GuiMain_); to provide these features (instead of calling GuiMain_() directly).

Subject: Re: Clean Exit
Posted by Mindtraveller on Sun, 27 May 2012 19:54:45 GMT
View Forum Message <> Reply to Message

Good news.
Exc is becoming even more useful.

Subject: Re: Clean Exit
Posted by nlneilson on Tue, 05 Jun 2012 00:31:44 GMT
View Forum Message <> Reply to Message

I had a problem closing apps that left threads still running.
The problem and how I solved it is in this thread:
  http://www.ultimatepp.org/forum/index.php?t=msg&goto=301 20&&srch=exit#msg_30120

Does the New function

void Exit(int code = 1);

do anything to close the threads or do they still need to be closed in a similar manner as in the link?

Subject: Re: Clean Exit
Posted by mirek on Tue, 05 Jun 2012 06:16:50 GMT

nlneilson wrote on Mon, 04 June 2012 20:31I had a problem closing apps that left threads still running.
The problem and how I solved it is in this thread:
   http://www.ultimatepp.org/forum/index.php?t=msg&goto=301 20&&srch=exit#msg_30120

Does the New function

void Exit(int code = 1);

do anything to close the threads or do they still need to be closed in a similar manner as in the link?

At this moment, it can only be invoked from the main thread and only terminates the main thread...

I would like this to work universally, but interrupting threads still seems to be tough issue...