Subject: Implementing a crash handler
Posted by crydev on Thu, 04 Dec 2014 07:55:07 GMT
View Forum Message <> Reply to Message

Hello,

I'm implementing a crash handler for my application. I used SetUnhandledExceptionFilter to let unhandled SEH exceptions flow into my own exception handler. My exception handler then executes a TopWindow with information about the crash. This works great when the exception is raised on the UI thread. The window may be created in the exception handler and it blocks until the window is closed.

I'm facing trouble with exceptions that are raised on a different thread. Using the Dispatch package from Bazaar (DispatcherCB) I tried to let the (asynchronous) exception handler dispatch the exception to the main window that uses PostCallback to invoke the window execution. Using an Atomic I tried to have the exception handler block until the window is closed but this only works partially.

With this solution, the next problems are:

When the exception is raised on the UI thread, the application enters an infinite loop, because the exception handler is blocking execution; When using EXCEPTION_EXECUTE_HANDLER to return from the exception handler, the application should terminate. Since I implemented the above described solution, the application does not terminate after the crash handler window is closed and destroyed.

I'm really having trouble with this. Is there an easier way to get the crash handler to be executed in a thread-safe way? Maybe there is an easier U++ builtin way?

Thanks a lot!

crydev