Subject: Re: thread, virtual class and exception
Posted by mrjt on Mon, 07 Apr 2008 10:53:37 GMT
View Forum Message <> Reply to Message

I'm afraid your example isn't terribly clear, but I think I can guess what you are trying to do.

You have a server, and various types of socket class. The server listens for a connection, creates the correct type of class to handle it and then runs a thread, passing the class object. These socket classes throw exceptions that you need to display on the GUI. Is this correct?

I think the main problem is that you are attempting to update the GUI from the thread. In Upp only the main thread (the server in this example) is allowed to do this. Also attempting to pass the messages up by exceptions is a bad idea, what you should do instead is pass a callback to a function in the main thread that can handle to GUI updates:
void fc(void *ss, Callback1<String> WhenException); // Thread function
void OnException(String exc); // Updates GUI
I'd also try and avoid creating objects with new, passing as void * and then recasting. you could for instance use the One<> container for this:

void fc(One<base_class> obj, Callback1<String> WhenException); // Thread function
..
One<base_class> obj;
obj = new inheritor(a, b, c);
thread.Run(THISBACK2(fc, obj, THISBACK(OnException));
This avoids the heap-leak as the One container takes ownership of the pointer and deletes it when the thread finishes.
One reason this may not have been working before is that you missed the virtual keyword from the vt() function decl. in the clss class.

Hope that helps, I rewrote your example to check that this all works so I've attached my test if it's any use.

James

File Attachments

1) test.zip, downloaded 301 times