Subject: Re: how to stop a thread that is waiting, e.g., listen()
Posted by tvanriper on Thu, 27 Sep 2007 11:43:12 GMT
View Forum Message <> Reply to Message

Out of my own curiosity, I took a peek at Thread.h/Thread.cpp in Core, and socket.h/socket.cpp in Web.

Thread.h/.cpp detail the Thread class, a class you can use for handling multi-threaded needs in an application. It exposes a Wait function, which returns an int. The integer it returns is either the exit code for the thread, or a numeric identifier for an event that was sent to the thread (well... in Windows... in POSIX, it seems to work differently, if I am reading this correctly... pthread_join [which will return 0] or the return value of the terminated thread).

For Windows, the Socket class seems to support the use of the Event object, but that object is not exposed to POSIX, if I'm reading everything correctly.

At a guess, you could probably create a Socket, set it up for networking however you need to, and use the static 'Wait' command with a timeout to wait on network traffic. If there's traffic, do what you need to do with the traffic. Otherwise, check a variable shared between your threads to see if you need to shut down your thread.

Perhaps someone else might have a better way to do this, but this seems to be the safest way to handle Sockets, multi-threading, and keeping your application working in both Windows and POSIX environments.