

## Re: How do I tell an object to free up an owned object from thta object itself?

*Source:* <http://coding.derkeiler.com/Archive/Delphi/comp.lang.pascal.delphi.misc/2003-10/0108.html>

---

*From:* Bo Berglund ([bo.berglund\\_at\\_telia.com](mailto:bo.berglund_at_telia.com))

*Date:* 10/03/03

Date: Fri, 03 Oct 2003 05:55:18 GMT

Thanks for your extensive analysis. I have been away a few days now for testing and I have the following conclusions:

A) Yes, it is possible to call `Self.Free` from within the handler object itself in the `SocketEvent` method if a disconnect is detected.

B) But I cannot do the `FSocket.Free` in this destructor. It causes an access violation. So I left that for the `TServerSocket` to handle.

C) I tested running a sequence of connect/disconnect from the client (clean, without forcing the application to exit). I had added a function to the server object to count the connections and list the client address for those connections that were active. Results:

- The connections list accumulates over time. It does not look like the `TServerSocket` frees up sockets that are no longer in use.
- The connections associated with clients that were forcibly closed (for example in Delphi debugger by stopping the program) are listed with active connections.

D) So I had to add a server side timer that every 15 seconds runs through the connections list and frees all sockets that are not connected.

E) A strange thing is the following:

If I stop listening (`TServerSocket.Close`) then all client sockets are disconnected. I can thus not have the server listen for a given time and allow connections, then stop listening but still let the already connected clients continue communicating. Strange, this makes it more difficult to set an upper limit on the number of clients allowed.

Finally, referring to one of your items:

No, I don't keep a list of created handlers in the server. They are just created in the `OnClientConnect` and are allowed to live independent lives. But the owner is the server object and they are descendants of `TComponent` so behind the scenes I guess some sort of list is kept nevertheless....

comp.lang.pascal.delphi.misc: Re: How do I tell an object to free up an owned object from thta object itself?

/Bo

On Wed, 01 Oct 2003 09:35:19 GMT, "David Reeve"  
<dree4456@bigpond.net.au> wrote:

>  
>"Bo Berglund" <bo.berglund@telia.com> wrote in message  
>news:krdinvgvlhp8ihr58p1vrflfjfpqv218od@4ax.com...  
>> On Tue, 30 Sep 2003 04:38:28 GMT, "David Reeve"  
>> <dree4456@bigpond.net.au> wrote:  
>>  
> [snip]  
>  
>I've been trying to get a good understanding of ScktComp.pas, so have done  
>a few tests on what you say here. I added a copy of ScktComp to my tester  
>so I could step it. With all those similarly named objects, it is very easy  
>to end up completely fuddled :-)  
>  
>  
>> Actually, what I am doing is this:  
>> 0. I have a Server object that creates a TServerSocket for listening  
>> 1. I set it to NonBlocking (that is what I am used to)  
>  
>OK... the OnClientxxxxxx events will be enabled in this mode.  
>  
>> 2. In the ClientConnect event I create a handler object for processing  
>> 3. The handler object is owned by the Server  
>> 4. The handler has a TCustomWinSocket variable that gets assigned  
>> the Socket supplied in the ClientConnect event  
>  
>Fine.... the object supplied in the OnClientConnect event that is called  
>Socket is in fact a TServerClientWinSocket which acts as a end-connection  
>for one client. What happens if a second client connection is made before  
>the first is disconnected? I assume you instantiate a new handler object and  
>associate the new 'Socket' object with that. So I guess you are managing a  
>list of handler objects at the TServerLevel?

No list is kept

>  
>> 5. All client processing hencefort is done by the handler  
>> 6. Then the client disconnects and I can free internal handler objects  
>  
>There is a possible catch here. The TServerClientWinSocket associated with  
>the client-connection is freed asynchronously by posting a message to itself  
>\*after\* your OnClientDisconnect handler has run. (All TServerClientWinSocket  
>instances are provided with a window and message handling courtesy of the  
>TCustomWinSocket ancestor, and respond to messages arrising from the sockets  
>API).

Re: How do I tell an object to free up an owned object from thta object itself?

comp.lang.pascal.delphi.misc: Re: How do I tell an object to free up an owned object from thta object itself?

My tests indicate the client sockets are not automatically freed until the server application itself is exited.

>> 7. But in order not to fill up memory I also need to dispose of two  
>> more items:  
>  
>Two more items? Aren't these freed at point 6?  
>  
>> –The handler itself (which I created in the ClientConnect event)  
>  
>This is simply a matter of looking through your list of handlers, and  
>removing, and freeing the handler associated with the 'Socket' object passed  
>in the OnDisconnect event.

The OnDisconnect never happens, instead there is a SocketEvent in the client handler socket and this whole discussion in a way is caused by this. I get the disconnect info on the handler level, not the server level.

>  
>> –The Socket that was supplied in the ClientConnect event.  
>  
>This done for you asynchronously after the OnDisconnect code has run..... as  
>discussed above.

No automatic socket disposal at all....

>  
>> Apparently the TServerSocket does not know of some client disconnects  
>> and thus the socket lingers on. For example if the client program is  
>> shut down unexpectedly then the TServerSocket still lists the socket  
>> as active. I don't think that the socket is ever destroyed unless I  
>> can initiate some cleanup action.  
>  
>I can't fault TServer socket in my test. I have verified that multiple  
>connections add to the connection list, and are removed as expected,  
>regardless of whether you call close on the clients, or pre-emptively close  
>them. However, this simple test may not cover all eventualities. Memproof  
>confirms that cleanup proceeds as expected.

Only clanup I can see is when I close the server program itself.

>  
>>  
>> My program will be running as an NT service in the end and thus it  
>> will be active a long time with many clients connecting/disconnecting  
>> over time.  
>> I don't want memory or socket usage to accumulate over time....  
>>  
>  
>I can certainly see the possibility for a problem, and you would need to  
>know the failure rate if you intended to run the service over a long period

Re: How do I tell an object to free up an owned object from thta object itself?

comp.lang.pascal.delphi.misc: Re: How do I tell an object to free up an owned object from thta object itself?

>of time. However, I would work toward getting a low disconnect failure rate  
>before implementing a more 'belt&braces' approach of forcing a time-to-live  
>on every client-connection.

I introduced a cleanup timer that removes non-used sockets. No limit to  
time of usage though, if connected they stay connected.

>  
>Dave  
>  
>

Bo Berglund  
bo.berglund@telia.com