Subject: Re: [CODE] cScTextObj class definition
Posted by ThisLittleGirl on Sat, 07 Apr 2012 20:58:10 GMT
View Forum Message <> Reply to Message

iRANian wrote on Sat, 07 April 2012 13:13The The RENEGADE_FUNCTION macro is a define for __declspec naked,
Ah, thought so.

iRANian wrote on Sat, 07 April 2012 13:13if you try to define the Constructor() function as a normal constructor (i.e. cScTextObj:cScTextObj()) the linker gives annoying errors.
Perhaps the NetworkObjectClass you're deriving from (or a base class of it) has its constructor declared private and/or left undefined so you can't actually instantiate it from your code. Because __declspec (naked) works for me on constructor definitions as well. It would in some way make sense do disallow construction within your code since all the code is within renegade. And you're not actually constructing the object yourself, renegade's Constructor function does it. Construction on pre-allocated memory is done with placement-new, something like:

new (TextObj) cScTextObj;

The many faces of new, haha. I hate it, hardly ever even use explicit new/delete since C++11.

iRANian wrote on Sat, 07 April 2012 13:13
Creating the object on the stack causes the FDS to freeze or crash on scope exit (not sure why this is),

Hmm, weird, that means you can construct it though. But you'll still have to call your Constructor function because your actual constructor will do/does nothing.

Have you thought about the possibility that renegades code may actually write beyond the class' size (as in, you didn't declare all members)? Does it even write into the object? If so writing past it on the stack almost guarantees a crash, haha.

BTW, where can I get to know more about this FDS? Ty.