
Subject: Re: [CODE] cScTextObj class definition
Posted by [ThisLittleGirl](#) on Sat, 07 Apr 2012 18:57:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Just wondering, there's probably good reasons you do it this way and I may be missing something badly but...

```
iRANian wrote on Fri, 06 April 2012 06:14
RENEGADE_FUNCTION
cScTextObj* __thiscall cScTextObj::Constructor()
AT2(0x004B9140, 0x004B9140);
//AT2(0x004B5AA0, 0x004B5AA0);
```

Why not just declare the constructor as such instead of as a member function, like so:

```
RENEGADE_FUNCTION
cScTextObj::cScTextObj()
AT2(0x004B9140, 0x004B9140);
Wouldn't that work? The __thiscall should be redundant anyway because it's the default for
member functions on MSVC++. And what exactly does the RENEGADE_FUNCTION macro do?
```

iRANian wrote on Fri, 06 April 2012 06:14

```
// Create a new cScTextObj
cScTextObj* TextObj = (cScTextObj*)operator new(sizeof(cScTextObj));
TextObj = TextObj->Constructor();
// ...
delete TextObj;
```

Also you could just put the thing on the stack instead of that tedious allocation. And isn't what the Constructor function would return the this pointer that you passed in (implicitly via ecx (thiscall)) anyways? In that case the assignment is redundant. If it wasn't the case you'd certainly fuck up yourself later by deleting a bogus pointer.

So, to sum it up. If you declare the constructor as such, you could just do:

```
cScTextObj text_obj;
```

Just curious, haha. ;p