

---

Subject: [CODE] cScTextObj class definition

Posted by [iRANian](#) on Fri, 06 Apr 2012 13:14:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

This is what the engine uses to send out most text "stuff" to players. It's used by the PPAGE, MSG, AMMSG, TMSG console commands and it's called whenever a player sends team, public and private chat to the server (to echo the text to the target(s)).

ID is the player ID of the guy to send the text as, this is set to -1 if you want to send as the host (for PPAGE, MSG etc)

Target is a specific player ID to send the message to, leave this -1 if you don't have a specific target (e.g. want to send a team chat or a HOST message).

Type can be public, private or team chat. Team chat is done with an 'ID' that is a valid player ID, 'Target' set to -1 and type set to TEXT\_MESSAGE\_TEAM (value is 1).

Message is the message to send..

IsPopup is used by the AMMSG console command and variants of it, it will show the text as a message window.

.h

```
class cScTextObj : public NetworkObjectClass
```

```
{
```

```
public:
```

```
int ID;
```

```
int Target;
```

```
TextMessageEnum Type;
```

```
WideStringClass Message;
```

```
bool IsPopup;
```

```
void Init(const WideStringClass& Message, TextMessageEnum Type, bool IsPopup, int ID, int Target);
```

```
cScTextObj* Constructor();
```

```
void Act();
```

```
}
```

.cpp (without the header #includes)

```
void* HookupAT3x(void* a, void* b, void* c, void* patch_start, void* patch_end, int (*version_selector)())
```

```
{
```

```
return HookupAT3(a,b,c,patch_start,patch_end,version_selector);
```

```
}
```

```
RENEGADE_FUNCTION
uint Send_Object_Update(NetworkObjectClass* object, int remoteHostId)
AT2(0x00461820,0x004612F0);
```

```
RENEGADE_FUNCTION
void cScTextObj::Act()
AT2(0x004B9720, 0x004B9720);
```

```
RENEGADE_FUNCTION
cScTextObj* __thiscall cScTextObj::Constructor()
AT2(0x004B9140, 0x004B9140);
//AT2(0x004B5AA0, 0x004B5AA0);
```

```
RENEGADE_FUNCTION
void cScTextObj::Init(const WideStringClass& Message, TextMessageEnum Type, bool IsPopup,
int ID, int Target)
AT2(0x004B91F0, 0x004B91F0);
```

Examples:

Have PlayerID show him sending a team chat message with the text "DERP", only on his client (the other players won't see this team chat message)

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

```
TextObj->ID = PlayerID; // Sender is PlayerID
TextObj->Type = TEXT_MESSAGE_TEAM; // Team chat message
TextObj->Message = L"DERP"; // Message to send
```

```
// Show the message only to PlayerID
TextObj->Set_Object_Dirty_Bits(PlayerID, NetworkObjectClass::BIT_CREATION);
Send_Object_Update(TextObj, PlayerID);
```

```
// Delete stuff
TextObj->Set_Delete_Pending();
delete TextObj;
```

Show a message window for every player (same thing as AMSG does):

```
cScTextObj* TextObj = (cScTextObj*)operator new(sizeof(cScTextObj));
TextObj = TextObj->Constructor();
TextObj->Init(L"DERP", TEXT_MESSAGE_PUBLIC, true, -1, -1);
```

Send a host message to all players not running scripts 2.9 or higher and send a white coloured message for those who do (can be expanded on to change the name of a player and send the text as that player, then change his name back):

```
void Send_Special_Host_Message(const char *Format, ...)
{
    char buffer[256];

    va_list va;
    _crt_va_start(va, Format);
    vsnprintf(buffer, 256, Format, va);
    va_end(va);

    cScTextObj* TextObj = (cScTextObj*)operator new(sizeof(cScTextObj));
    TextObj = TextObj->Constructor();

    TextObj->Message = buffer;

    for (SLNode<cPlayer>* PlayerIter = Get_Player_List()->Head(); (PlayerIter != NULL); PlayerIter =
    PlayerIter->Next())
    {
        cPlayer *p = PlayerIter->Data();

        if (p->IsActive && Get_GameObj(p->Get_Id()))
        {
            float Version = Get_Client_Version(p->Get_Id());

            if (Version < 2.9)
            {
                TextObj->Set_Object_Dirty_Bits(p->Get_Id(), NetworkObjectClass::BIT_CREATION);
                Send_Object_Update(TextObj, p->Get_Id());
            }
            else
            {
                Send_Message_Player(Get_GameObj(p->Get_Id()), 250, 255, 255, buffer);
            }
        }
    }
    TextObj->Set_Delete_Pending();
    delete TextObj;
}
```

Thanks to StealthEye with helping me out with getting this to work and showing that intermixing CMSGP and MSG can be done. I got lots of useful information for reversing this from the OnOeS source code (the cCsTextObj hooking stuff).