
Subject: [CODE] SCAnnouncement class definition
Posted by [iRANian](#) on Thu, 05 Apr 2012 15:52:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is what the engine uses to send out radio commands to players after a player sends a radio command to the server. If an invalid player ID (e.g. ID = -1) is used, the radio command will be shown as if it were an announcement, which basically means that 'AnnouncementID' (an ID from strings.tdb) will be shown in a white colour and if it has a sound ID it will be played, the same as with normal radio commands, but with a white colour and with no player name prepended to it.

IconID needs to be between 0 or 1 and 30 or 31 or the radio command won't be sent.

PlayerType is the team of the player, this appears to be ignored.

AnnouncementType seems to be ignored too, no idea what it's for.

AnnouncementID is an ID from strings.tdb, it's the string ID to be displayed and if the strings.tdb ID has a sound ID attached it'll also play a sound (not for all strings, probably because the sound file is missing).

ID is the player ID of the player sending the message.

.h

```
class SCAnnouncement : public NetworkObjectClass
{
public:

    int PlayerType;
    int ID;
    int AnnouncementID;
    int IconId;
    uint8 AnnouncementType;

    SCAnnouncement* Constructor();

    void Init(int PlayerType, int ID, int AnnouncementID, uint8 AnnouncementType, int IconId );
    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
```

```
SCAnnouncement* __thiscall SCAnnouncement::Constructor()  
AT2(0x004B38B0, 0x004B38B0);
```

```
RENEGADE_FUNCTION
```

```
void SCAnnouncement::Init(int PlayerType, int ID, int AnnouncementID, uint8  
AnnouncementType, int IconId )  
AT2(0x004B3930, 0x004B3930);
```

```
RENEGADE_FUNCTION
```

```
void SCAnnouncement::Act()  
AT2(0x004B3D10, 0x004B3D10);
```

Example usage:

Code emulating PlayerID sending the ALT+3 (move out) radio command:

```
SCAnnouncement* RadioObj = (SCAnnouncement*)operator new(sizeof(SCAnnouncement));  
RadioObj = RadioObj->Constructor();  
RadioObj->Init(Get_Team(PlayerID), PlayerID, 8547, 1, 12);
```

Show a white coloured message "You have been banned from Westwood Online." for one player:

```
SCAnnouncement* RadioObj = (SCAnnouncement*)operator new(sizeof(SCAnnouncement));  
RadioObj = RadioObj->Constructor();  
  
RadioObj->PlayerType = 0; // Doesn't seem to be checked  
RadioObj->ID = -1; // Invalid player ID, which will cause the radio command to be shown in white  
without a sender  
RadioObj->IconId = 12; // IconId is the same as for the ALT+3 ("Move out.") radio command.  
RadioObj->AnnouncementType = 1; // Doesn't seem to be used  
RadioObj->AnnouncementID = 9757; // strings.tdb ID for "You have been banned from  
Westwood Online."  
  
// Sends the radio command only to PlayerID.  
RadioObj->Set_Object_Dirty_Bits(PlayerID, NetworkObjectClass::BIT_CREATION); // Netcode  
stuff  
Send_Object_Update(RadioObj, PlayerID); // Send an object update for only this player  
  
RadioObj->Set_Delete_Pending();  
delete RadioObj;
```

Subject: Re: [CODE] SCAnnouncement class definition
Posted by [sla.ro\(master\)](#) on Thu, 05 Apr 2012 16:26:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

nice code. i was finding a code who can display models only for a player

Subject: Re: [CODE] SCAnnouncement class definition
Posted by [halo2pac](#) on Fri, 06 Apr 2012 00:00:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

I wonder what else you can do with Send_Object_Update and NetworkObjectClass

Subject: Re: [CODE] SCAnnouncement class definition
Posted by [iRANian](#) on Fri, 06 Apr 2012 00:23:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

halo2pac wrote on Thu, 05 April 2012 17:00I wonder what else you can do with
Send_Object_Update and NetworkObjectClass

A lot of stuff, like sending cScTextObj to only one player, and creating powerups that can only be
picked up by one player, delete crate objects only for people who have an SBH, sending radio
commands to both teams.

Subject: Re: [CODE] SCAnnouncement class definition
Posted by [halo2pac](#) on Fri, 06 Apr 2012 18:58:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wait... how many years has this stuff been around?

Subject: Re: [CODE] SCAnnouncement class definition
Posted by [iRANian](#) on Fri, 06 Apr 2012 20:14:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Those functions/classes are part of the engine.

Subject: Re: [CODE] SCAnnouncement class definition
Posted by [halo2pac](#) on Mon, 09 Apr 2012 04:20:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm confused, if they are part of the engine... why are you showing them here? Just so you can
make people aware of them or is it because your doing something special with the code that is
beyond the default?

Subject: Re: [CODE] SCAnnouncement class definition

Posted by [jonwil](#) on Mon, 09 Apr 2012 04:39:58 GMT

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

He is showing them so that people have the information required to call them and make use of them.

Subject: Re: [CODE] SCAnnouncement class definition

Posted by [jonwil](#) on Mon, 09 Apr 2012 17:13:43 GMT

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

I have added a new engine call to 4.0 beta 5 as follows:

```
SCAnnouncement *Send_Client_Announcement(int _receiptentId, int _senderId, int  
_translationId, AnnouncementEnum _type, int _emoticonId, bool dodirtybit, bool doact)
```

dodirtybit determines whether SCAnnouncement::Init should make its normal calls to the various Set_Object_Dirty_Bit functions or not

doact determines whether SCAnnouncement::Init should make its normal calls to SCAnnouncement::Act

The function will create a new SCAnnouncement, call its constructor, call SCAnnouncement::Init and return the new object.

Note that you should not call Set_Delete_Pending, nor should you delete the returned object as Set_Delete_Pending is already handled by SCAnnouncement::Init and the object will be automatically deleted by the netcode.

You are free to call any of the Set_Object_Dirty_Bit functions and to call Send_Object_Update. Note however that you can only call Send_Object_Update ONCE for each SCAnnouncement object as SCAnnouncement was not designed to be sent multiple times.

The purpose of the engine call is to make this stuff easier to use and to make sure it wont break in the future if SCAnnouncement has to change for some reason.

Subject: Re: [CODE] SCAnnouncement class definition

Posted by [Whitedragon](#) on Mon, 09 Apr 2012 23:05:20 GMT

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

halo2pac wrote on Sun, 08 April 2012 21:20 I'm confused, if they are part of the engine... why are you showing them here? Just so you can make people aware of them or is it because your doing something special with the code that is beyond the default?

This displays a string from strings.tdb and plays the associated sound. It's a good way to send certain messages to players who aren't running tt.dll. Of course the string needs to be in the strings database but there's quite a bit of stuff in there.

```
void Send_Announcement_Player(int ID,int StringID) {
  SCAnnouncement *RadioEvent = (SCAnnouncement*)operator new(sizeof(SCAnnouncement));
  RadioEvent->Constructor();
  RadioEvent->AnnouncementID = StringID;
  RadioEvent->ID = -1;
  RadioEvent->IconID = 0;
  RadioEvent->AnnouncementType = 0;
  RadioEvent->PlayerType = 0;
  RadioEvent->Set_Object_Dirty_Bits(ID,NetworkObjectClass::BIT_CREATION);
  Send_Object_Update(RadioEvent,ID);
  RadioEvent->Set_Delete_Pending();
  delete RadioEvent;
}
```

```
Send_Announcement_Player(ID,1496);
```

File Attachments

1) [Screenshot.7.png](#), downloaded 853 times

