

---

Subject: old ssgm chatcommandclass ported to 4.0  
Posted by [robbyke](#) on Fri, 30 Mar 2012 13:38:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

ive remade as i thought it was easy to use ill just share if anyone else wants it

CommmandClass.h

```
/* Renegade Scripts.dll  
SSGM chat command classes and functions  
Copyright 2007 Whitedragon(MDB), Jonathan Wilson
```

```
This file is part of the Renegade scripts.dll  
The Renegade scripts.dll is free software; you can redistribute it and/or modify it under  
the terms of the GNU General Public License as published by the Free  
Software Foundation; either version 2, or (at your option) any later  
version. See the file COPYING for more details.  
In addition, an exemption is given to allow Run Time Dynamic Linking of this code with any  
closed source module that does not contain code covered by this licence.  
Only the source code to the module(s) containing the licenced code has to be released.
```

```
*/  
#include "Defines.h"  
#include "Standard.h"  
#include "HashTemplateClass.h"  
#include "HashTemplateliterator.h"  
  
#ifndef COMMANDS_H  
#define COMMANDS_H  
  
class TokenClass {  
private:  
DynamicVectorClass<StringClass> Tokens;  
int vecsize;  
void Build(const StringClass &Text,int Pos) {  
Tokens.Clear();  
vecsize = 0;  
char *Tokenz = new char[Text.Get_Length()+1];  
sprintf(Tokenz,"%s",Text);  
char *p = strtok(Tokenz," ");  
StringClass Temp2,All;  
if (!Pos) {  
Tokens.Add(Text);  
}  
else {  
int i = 0;  
while (i < Pos) {  
p = strtok(0," ");
```

```

    ++i;
}
}
while (p) {
    Temp2 = p;
    Tokens.Add(Temp2);
    p = strtok(0, " ");
    ++vecsize;
    if (Pos) {
        All += Temp2;
        if (p) All += StringClass(" ");
    }
}
if (Pos) {
    Tokens.Insert(0,All);
}
delete[] Tokenz;
}

```

public:

```

TokenClass(const TokenClass &Copy) {
    Tokens = Copy.Tokens;
    vecsize = Copy.vecsize;
}
TokenClass() { }
TokenClass(const StringClass &Text,int Pos = 0) {
    Build(Text,Pos);
}

```

```

TokenClass& operator=(const TokenClass &Copy) {
    Tokens = Copy.Tokens;
    vecsize = Copy.vecsize;
    return *this;
}

```

```

TokenClass& operator=(const StringClass &Text) {
    Build(Text,0);
    return *this;
}

```

```

inline StringClass operator[](int Pos) const {
    if (vecsize < Pos) {
        return "";
    }
    return Tokens[Pos];
}

```

```

StringClass operator()(int Start,int End = 0) const {
if (vecsize < Start || vecsize < End) {
return "";
}
StringClass Ret;
if (!End) {
End = Tokens.Length();
}
int i = Start;
while (i <= End && i <= vecsize) {
Ret += Tokens[i];
++i;
if (i <= End) Ret += StringClass(" ");
}
return Ret;
}

inline int size() const {
return vecsize;
}

inline void erase(int Pos) {
if (vecsize < Pos) return;
Tokens.Delete(0+Pos);
vecsize--;
}

inline void replace(int Pos,const StringClass &Rep) {
if (vecsize < Pos || !Pos) return;
Tokens[Pos] = Rep;;
}

inline void eraseglobal(int Pos) {
if (vecsize < Pos) return;
StringClass Temp = Tokens[0];
Temp.crop(Temp.Format(Tokens[Pos]),Tokens[Pos].Get_Length()+1);
Tokens[0] = Temp;
erase(Pos);
}

inline void Add(const StringClass &Text,int Pos = 0) {
if (!Pos) {
Tokens.Add(Text);
++vecsize;
}
else if (vecsize < Pos) {
return;
}
}

```

```

else {
    Tokens.Insert(0+Pos,Text);
    ++veclsize;
}
}
};

struct ChatCommandInfo;

class ChatCommandClass {
public:

    ChatCommandInfo *Info;
    virtual void Error(int ID,int ErrorType,int Param);
    virtual void Triggered(int ID,const TokenClass &Text) = 0;
};

class ChatCommandList {
public:
    static HashTemplateClass<StringClass,ChatCommandInfo *> CommandsMap;
    static void Add_Chat_Command(ChatCommandClass *Ptr,const char *Command,int
NumParams);
};

struct ChatCommandInfo {
    StringClass Command;
    ChatCommandClass *Ptr;
    int NumParams;
    int GameMode;
};

template <class T> class ChatCommandRegistrant : public ChatCommandList {
public:
    ChatCommandRegistrant(const char *Command,int NumParams) {
        char *Comm = newstr(Command);
        char *p = strtok(Comm,";");
        while (p) {
            ChatCommandClass *Temp = new T;
            Add_Chat_Command(Temp,p,NumParams);
            p = strtok(0,";");
        }
        delete[] Comm;
    }
};

#endif

```

CommandClass.cpp

```
#include "Defines.h"  
#include "Standard.h"
```

```
HashTemplateClass<StringClass,ChatCommandInfo *> ChatCommandList::CommandsMap;
```

```
void ChatCommandClass::Error(int ID,int ErrorType,int Param) {  
    if (ID) {  
        if (ErrorType == 1) {  
            StringClass Msg;  
            Msg.Format("page %d Insufficient parameters. This command requires atleast %d  
parameter(s), you only supplied %d.",ID,Info->NumParams,Param);  
            Console_Input(Msg);  
        }  
    }  
}
```

```
void ChatCommandList::Add_Chat_Command(ChatCommandClass *Ptr,const char  
*Command,int NumParams) {  
    ChatCommandInfo *Temp = new ChatCommandInfo;  
    Temp->Ptr = Ptr;  
    Temp->Command = Command;  
    Temp->NumParams = NumParams;  
    Ptr->Info = Temp;  
    ChatCommandList::CommandsMap.Insert(Command,Temp);  
}
```

example command

```
class ChatCommandExample : public ChatCommandClass  
{  
    void Triggered(int ID,const TokenClass &Text)  
    {  
        //Do some shit here  
    }  
};  
ChatCommandRegistrant<ChatCommandExample>  
ChatCommandExample_Reg("!example;!ex",0);
```

how to make it work

```
bool Example::OnChat(int PlayerID,TextMessageEnum Type,const wchar_t *Message,int
recieverID)
{
    StringClass Msg = Message;
    if (Msg[0] == '!') {
        TokenClass Text(Msg);
        StringClass Command = Text[1];
        Text.erase(1);

        if (ChatCommandList::CommandsMap.Exists(Command))
        { // If so
            ChatCommandInfo *Temp = ChatCommandList::CommandsMap.Get(Command,0);
            Temp->Ptr->Triggered(PlayerID,Text);
        }
    }
    return true;
}
```

If i missed something pls tell me i just fast copied it out my plugin so might have made some errors

note:

i removed chattype out of triggered because i didnt need it, it stil exist in the regristrant though if you want to use it its just a small change

EDIT : changed some stuff

I recommend using these files in your plugin.

---