
Subject: Re: old tokenclass

Posted by [robbyke](#) on Wed, 05 Oct 2011 06:35:06 GMT

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

im running 4.0 im trying to convert my mod to 4.0 but lots of functions are different or dissappeared

ive been trying to convert the old command class but ive noticed that the new vector class doesnt has a function to determine its bgin. now im not dumb (i hope) and begin = 0. so i just use 0 but on the internet i found that if my vector is empty ill get an exeption bcause vector[0] is invalid

this is what ive become after converting

Chatcommandclass.h

Toggle Spoiler

```
/* 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 "General.h"
```

```
#ifndef COMMANDS_H
#define COMMANDS_H
```

```
#define CHATTYPE_PUBLIC 0
#define CHATTYPE_TEAM 1
#define CHATTYPE_ALL 2
```

```
#define GAMEMODE_ALL 0
#define GAMEMODE_AOW 1
#define GAMEMODE_CTF 2
#define GAMEMODE_SNIPIING 3
#define GAMEMODE_500SNIPIING 4
#define GAMEMODE_INFANTRY 5
```

```
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);
++vecsize;
}
}
};

```

```

struct DataStruct;
struct ChatCommandInfo;

```

```

class ChatCommandClass {
public:
ChatCommandInfo *Info;
virtual void Error(int ID,int ErrorType,int Param);
virtual void Triggered(int ID,const TokenClass &Text,int ChatType) = 0;
};

```

```

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

```

```

class ChatCommandList {
public:
static DynamicVectorClass<ChatCommandInfo*> *List;
static void Add_Chat_Command(ChatCommandClass *Ptr,const char *Command,int
ChatType,int NumParams,int GameMode);
};

```

```

template <class T> class ChatCommandRegistrant : public ChatCommandList {

```

```

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

};

#endif

```

Chatcommandclass.cpp
Toggle Spoiler

```

/* 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 "General.h"
```

```

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);
}
}
}
}

```

```
DynamicVectorClass<ChatCommandInfo*> *ChatCommandList::List = 0;
```

```
void ChatCommandList::Add_Chat_Command(ChatCommandClass *Ptr,const char
*Command,int ChatType,int NumParams,int GameMode) {
    ChatCommandInfo *Temp = new ChatCommandInfo;
    Temp->Ptr = Ptr;
    Temp->Command = Command;
    Temp->ChatType = ChatType;
    Temp->NumParams = NumParams;
    Temp->GameMode = GameMode;
    Ptr->Info = Temp;
    if (!List) {
        List = new DynamicVectorClass<ChatCommandInfo*>;
    }
    List->Add(Temp);
}
```

ive just copied those out of the old ssgm and edited them
