Subject: Re: weird reaction with const w_chart
Posted by iRANian on Sun, 11 Mar 2012 16:03:15 GMT
View Forum Message <> Reply to Message

robbyke wrote on Sun, 11 March 2012 08:34i tried to convert it once but i didnt understand vector then(stil not)

so ye if you want to it would be nice
Alright, it should work the same way the stuff in SSGM 2.0.2 worked like (accessing [0] gives you the full string, [1] gives you the first token). Not sure if there's any issues with it.

Tokenizer.cpp:
Toggle Spoiler#include "Tokenizer.h"

```
void Tokenizer::Build(const StringClass &Text, int Pos)
{
 Tokens.Clear();
 VectorSize = 0;
 StringClass Temp2, All;

 StringClass Tokenz = Text;
 char *p = strtok(Tokenz.Peek_Buffer()," ");

 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, " ");
  ++VectorSize;

  if (Pos)
  {
   All += Temp2;
   if (p) All += " ";
```

```
  }
 }
 if (Pos)
 {
  Tokens.Add_Head(All);
 }
}

Tokenizer::Tokenizer(const Tokenizer &Copy)
{
 Tokens = Copy.Tokens;
 VectorSize = Copy.VectorSize;
}

Tokenizer::Tokenizer()
{
}

Tokenizer::Tokenizer(const StringClass &Text, int Pos)
{
 Build(Text, Pos);
}

Tokenizer& Tokenizer::operator=(const Tokenizer &Copy)
{
 Tokens = Copy.Tokens;
 VectorSize = Copy.VectorSize;
 return *this;
}

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

StringClass Tokenizer::operator[](int Pos)
{
 if (VectorSize < Pos)
 {
  return "";
 }
 return Tokens[Pos];
}

StringClass Tokenizer::operator()(int Start,int End)
{
 if (VectorSize < Start || VectorSize < End)
```

```
 {
  return "";
 }
 StringClass Ret;
 if (!End) {
  End = Tokens.Count();
 }
 int i = Start;

 while (i <= End && i <= VectorSize)
 {
  Ret += Tokens[i];
  ++i;
  if (i <= End)
   Ret += StringClass(" ");
 }
 return Ret;
}

int Tokenizer::Size()
{
 return VectorSize;
}

void Tokenizer::Erase(int Pos)
{
 if (VectorSize < Pos) return;

 Tokens.Delete(Pos);
 VectorSize--;
}

void Tokenizer::Replace(int Pos, const StringClass &Rep)
{
 if (VectorSize < Pos || !Pos) return;

 Tokens[Pos] = Rep;
}

 /* inline void Tokenizer::Erase_Global(int Pos)
{
 if (VectorSize < Pos) return;

 StringClass Temp = Tokens[0];
 Temp.Replace(Temp.find(Tokens[Pos]),Tokens[Pos].size()+1,"");
 Tokens[0] = Temp;
 Erase(Pos);
} */
```

```
inline void Tokenizer::Add(const StringClass &Text, int Pos)
{
 if (!Pos)
 {
  Tokens.Add(Text);
  ++VectorSize;
 }
 else if (VectorSize < Pos)
 {
  return;
 }
 else
 {
  Tokens.Insert(Pos, Text);
  ++VectorSize;
 }
}
```

Tokenizer.h:
Toggle Spoiler#pragma once

```
#include "gmplugin.h"

// First position in the Tokenizer string contains the full string, so use [1] instead of [0]
// for the first string

class Tokenizer
{
private:
 DynamicVectorClass<StringClass> Tokens;
 int VectorSize;
 void Build(const StringClass &Text, int Pos);

public:
 Tokenizer();
 Tokenizer(const Tokenizer &Copy);
 Tokenizer(const StringClass &Text, int Pos = 0);
 Tokenizer& operator=(const Tokenizer &Copy);
 Tokenizer& operator=(const StringClass &Text);
 StringClass operator[](int Pos);
 StringClass operator()(int Start, int End = 0);
 int Size();
 void Erase(int Pos);
 void Replace(int Pos,const StringClass &Rep);
// void Erase_Global(int Pos);
 void Add(const StringClass &Text, int Pos = 0);
};
```