Subject: Re: old tokenclass
Posted by Jerad2142 on Tue, 04 Oct 2011 21:31:56 GMT
View Forum Message <> Reply to Message

Copied directly from ssgm, if you wanted to use it without ssgm you should rename its calls so you don't end up stepping on someone elses' toes.

```
class TokenClass {
private:
 std::vector<std::string> Tokens;
 int vecsize;
 void Build(const std::string &Text,int Pos) {
  Tokens.clear();
  vecsize = 0;
  char *Tokenz = new char[Text.size()+1];
  sprintf(Tokenz,"%s",Text.c_str());
  char *p = strtok(Tokenz," ");
  std::string Temp2,All;
  if (!Pos) {
   Tokens.push_back(Text);
  }
  else {
   int i = 0;
   while (i < Pos) {
    p = strtok(0," ");
    ++i;
   }
  }
  while (p) {
   Temp2 = p;
   Tokens.push_back(Temp2);
   p = strtok(0," ");
   ++vecsize;
   if (Pos) {
    All += Temp2;
    if (p) All += std::string(" ");
   }
  }
  if (Pos) {
   Tokens.insert(Tokens.begin(),All);
  }
  delete[] Tokenz;
 }

 public:

 TokenClass(const TokenClass &Copy) {
  Tokens = Copy.Tokens;
  vecsize = Copy.vecsize;
```

```
}
TokenClass() { }
TokenClass(const std::string &Text,int Pos = 0) {
 Build(Text,Pos);
}

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

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

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

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

inline int size() const {
 return vecsize;
}

inline void erase(int Pos) {
 if (vecsize < Pos) return;
 Tokens.erase(Tokens.begin()+Pos);
```

```
  vecsize--;
 }

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

 inline void eraseglobal(int Pos) {
  if (vecsize < Pos) return;
  std::string Temp = Tokens[0];
  Temp.replace(Temp.find(Tokens[Pos]),Tokens[Pos].size()+1,"");
  Tokens[0] = Temp;
  erase(Pos);
 }

 inline void Add(const std::string &Text,int Pos = 0) {
  if (!Pos) {
   Tokens.push_back(Text);
   ++vecsize;
  }
  else if (vecsize < Pos) {
   return;
  }
  else {
   Tokens.insert(Tokens.begin()+Pos,Text);
   ++vecsize;
  }
 }
};
```