
Subject: Re: SetLineKeyValue - in memory:update2
Posted by [luoganda](#) on Sun, 30 Apr 2017 12:33:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Using SetLineKeyValue with stringstream:

-stringstream: a few funcs should be added to use this,chk down

If you read this, there was also original func update - chk original message

Usage:

-to speedup and data(lines) is/are known to be contiguous,then;

```
SetLineKeyValue(ss,"a key", "a value");
```

```
ss.GetLine(); //advance pos to next line,so next is found immediatelly
```

-when keys(data) is not known to be contiguous(butAreSearchable),then;

```
SetLineKeyValue(ss,"a key", "a value"); //just this
```

```
bool SetLineKeyValue(StringStream& ss,const String& key,const String& val,bool  
chkForEqualSign,dword *chkForExistanceOnly,String *prevBack){
```

```
if(ss.IsEof())return false;
```

```
register char *p=(char*)ss.PeekPtr();
```

```
int64 fpos=ss.GetPos();ss.Seek(0);
```

```
char *pst,*porg=p,*ssSt=(char*)ss.PeekPtr();
```

```
int pos,off,linenum;
```

```
ss.Seek(fpos);
```

```
if(prevBack)*prevBack="";
```

```
while(*p){
```

```
    fpos=ss.GetPos(); linenum=0;pos=0;pst=p;
```

```
while(*p){if(*p=='\r'&&*p=='\n'){linenum=p-pst/*p~lines-fpos*/;break;}if(*p=='\r'||*p=='\n'){linenum=p  
-pst/*p~lines-fpos*/;break;}if(*p=='=')pos=p-pst/*p~lines-fpos*/; ++p;}
```

```
if(*p!=0)++p;
```

```
if(linenum==0 && 0==(linenum=p-(*p?1:0)-pst/*~lines-fpos*/))continue;
```

```
if(linenum&&*pst=='='||linenum>=2&&pst[0]=='/'&&pst[1]=='/')continue;
```

```
if(chkForEqualSign&&!pos)continue;
```

```
if(linenum<key.GetCount()+chkForEqualSign?1:0)||0!=memcmp(pst,~key,key.GetCount()))contin  
ue;if(chkForEqualSign&&pst[key.GetCount()]!='=')continue;
```

```
if(prevBack&&linenum>fpos+/*key.GetCount()*/pos+1)*prevBack=String(pst+/*key.GetCount()*/po  
s+1/*lines.Mid(fpos+key.GetCount()+1*/,linenum-/*key.GetCount()*/pos+1));
```

```
if(chkForExistanceOnly){*chkForExistanceOnly=(dword)(int32)fpos;return true;}
```

```
fpos+=/*key.GetCount()*/pos+1;
```

```
off=/*key.GetCount()*/pos+1+val.GetCount()-linenum; //speedup a litle - don't insert/remove
```

twice

```
if(off<0)ss.Remove(abs(off),fpos); else if(off)ss.Insert('R',off,fpos);
```

```
*(char*)ss.GetIter(fpos-1)='='; memcpy((void*)ss.GetIter(fpos),~val,val.GetCount());
```

```

//if(off<0)slns. lines.Remove(fpos,abs(off)); else if(off)lines.Insert(fpos," ",off);
/*(char*)lines.Getlter(fpos-1)!=''; memcpy((void*)lines.Getlter(fpos),~val,val.GetCount());
return true;
}
return false;
}

```

StringStream added code for this - but should be revisited;
StringStream has Put methods but no Remove(Pos isTakenInConsideration), here they are - but
are not yet 100% tested
optimally it should be added between SetSize and Open

```

virtual void Insert(int ch,int num=1,int at=-1);
virtual void Insert(const String& str,int at=-1);
virtual void Remove(int num=1,int at=-1);
virtual const byte* Getlter(int at){ASSERT(at<GetSize());return buffer+at;}

```

//cpp files

```

void StringStream::Insert(int ch,int num,int at){
  Insert(String(ch,num),at);
}
void StringStream::Insert(const String& str,int at){
  if(at==-1)at=GetPos();
  ASSERT(at+str.GetCount()<=GetSize());
  SetReadMode(); //justToKnowWhichToUse...
  int64 npos=at<GetPos()?GetPos()+GetPos()-at:GetPos();
  data.Insert(at,str);
  buffer = (byte *) ~data; //<=this should be checked,may point to another unknown buffer
  //ptr = wrlim = buffer;
  rdlim = buffer + data.GetCount();
  Seek(npos);
}
void StringStream::Remove(int num,int at){
  if(at==-1)at=GetPos();
  ASSERT(at+num<=GetSize());
  SetReadMode(); //justToKnowWhichToUse...
  int64 npos=at<=GetPos()?GetPos()-num/*+GetPos()-at*/:GetPos();
  data.Remove(at,num);
  //memmove(buffer+at,buffer+at+num,n);
  buffer = (byte *) ~data;
  rdlim = buffer + data.GetCount();
  Seek(npos);
}

```