Subject: Suggestion for StringBuffer
Posted by rxantos on Sun, 19 Jul 2015 22:50:52 GMT
View Forum Message <> Reply to Message

Could it be possible to Add

```
void  Set(const String& s);
StringBuffer(const String& s){ Zero(); Set(s);}
```

to StringBuffer in uppsrc/Core/String.h
and

```
void StringBuffer::Set(const String& s)
{
 int len = s.GetLength();
 if(0 < len) {
  SetLength(len);
  memcpy(begin, s.Begin(), len);
 } else {
  Zero();
 }
}
```

in uppsrc/Core/String.cpp

Sometimes I need top create a string buffer from a constant string to edit in a copy without affecting the original.

Subject: Re: Suggestion for StringBuffer
Posted by mirek on Sat, 29 Aug 2015 05:36:02 GMT
View Forum Message <> Reply to Message

Hi,

after carefully considering this, I tentatively decided it is not a good idea:

a) by adding StringBuffer(const String& s) the action that is going to happen would be quite strangely defined. Consider

```
void Fn(const String& p) { StringBuffer b(p); }
```

vs

```
void Fn(String p) { StringBuffer b(p); }
```

I would say this would be a great source of accidental errors...

b) Set is in function almost identical to Cat

So, if you need to assign data now, you can use

```
String s;
StringBuffer b;
b.Cat(s);
```

I think this is sufficient.

Anyway, all of this is not final. But if I could be convinced we need anything new here, I would rather use

```
StringBuffer(const String& s, int); // standard way of saying 'deep copy'
void Set(const String& s) { Clear(); Cat(s); } // simpler implementation
```

Mirek
