
Subject: Serious bug in String comparison or weird feature?

Posted by [Werner](#) on Sat, 09 Feb 2008 10:13:31 GMT

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

If you have 2 U++ strings of different length but apart from that identical, U++ tells you that the smaller string is greater (>) than the bigger one.

This seems wrong to me, and indeed std::string behaves the other way round, namely as expected - at least by me.

Consider for example this small app:

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
GUI_APP_MAIN
```

```
{  
/*
```

```
// start of U++ String version
```

```
const String a = "AddFrame";
```

```
const String b = "AddFrameSize";
```

```
String less = a < b ? "true" : "false";
```

```
String equal = a == b ? "true" : "false";
```

```
String greater = a > b ? "true" : "false";
```

```
PromptOK(Format("%s-<-|%s:|%s&%s|=|=|%s:|%s&%s|>-|%s:|%s", a, b, less, a, b, equal, a,  
b, greater));
```

```
// end of U++ String version
```

```
*/
```

```
// start of std::string version
```

```
const std::string a = "AddFrame";
```

```
const std::string b = "AddFrameSize";
```

```
String less = a < b ? "true" : "false";
```

```
String equal = a == b ? "true" : "false";
```

```
String greater = a > b ? "true" : "false";
```

```
String au(a.c_str());
```

```
String bu(b.c_str());
```

```
PromptOK(Format("%s-<-|%s:|%s&%s|=|=|%s:|%s&%s|>-|%s:|%s", au, bu, less, au, bu,  
equal, au, bu, greater));
```

```
// end of std::string version
```

```
}
```

By the way, you can easily fix that by adding a space to either of the U++ Strings.

Werner

Subject: Re: Serious bug in String comparison or weird feature?

Posted by [mirek](#) on Sun, 10 Feb 2008 08:02:49 GMT

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

Werner wrote on Sat, 09 February 2008 05:13: If you have 2 U++ strings of different length but apart from that identical, U++ tells you that the smaller string is greater (>) than the bigger one.

This seems wrong to me, and indeed std::string behaves the other way round, namely as expected - at least by me.

Consider for example this small app:

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
GUI_APP_MAIN
```

```
{
/*
// start of U++ String version
const String a = "AddFrame";
const String b = "AddFrameSize";

String less = a < b ? "true" : "false";
String equal = a == b ? "true" : "false";
String greater = a > b ? "true" : "false";
PromptOK(Format("%s-<-%s:-%s&%s-|=|-%s:-%s&%s->-%s:-%s", a, b, less, a, b, equal, a,
b, greater));
// end of U++ String version
*/
```

```
// start of std::string version
const std::string a = "AddFrame";
const std::string b = "AddFrameSize";
```

```
String less = a < b ? "true" : "false";
String equal = a == b ? "true" : "false";
String greater = a > b ? "true" : "false";
String au(a.c_str());
String bu(b.c_str());
PromptOK(Format("%s-<-%s:-%s&%s-|=|-%s:-%s&%s->-%s:-%s", au, bu, less, au, bu,
equal, au, bu, greater));
// end of std::string version
}
```

By the way, you can easily fix that by adding a space to either of the U++ Strings.

Werner

Ooops. Thanks. Fixed.

Quick fix:

```
template <class B>
int AString<B>::Compare(const String& s) const
{
    const tchar *a = B::Begin();
    const tchar *ae = End();
    const tchar *b = s.Begin();
    const tchar *be = s.End();
    for(;;) {
        if(a >= ae)
            return b >= be ? 0 : -1;
        if(b >= be)
            return 1;
        int q = cmpval__(*a++) - cmpval__(*b++);
        if(q)
            return q;
    }
}
```

```
template <class B>
int AString<B>::Compare(const char *b) const
{
    const tchar *a = B::Begin();
    const tchar *ae = End();
    for(;;) {
        if(a >= ae)
            return *b == 0 ? 0 : -1;
        if(*b == 0)
            return 1;
        int q = cmpval__(*a++) - cmpval__(*b++);
        if(q)
            return q;
    }
}
```

Mirek
