Subject: Compiling U++ Core with -Wall (GCC 4.1.2 Kubuntu 6.10)
Posted by mr_ped on Thu, 18 Oct 2007 19:53:39 GMT
View Forum Message <> Reply to Message

There's lot of warnings making the output of compilation too much cluttered, thus hard to spot warnings in my own code.

I think lot of them can be easily removed without risk of breaking current Core functionality.

But I'm not going to fix them, there's no point without serious version controlling system where I can commit changes directly.

but non-virtual destructor

/home/ped/upp/uppsrc/Core/Value.h:132: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:133: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:203: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:204: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:205: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h:205: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:206: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h:206: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h:207: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h:207: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:208: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h:208: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h: At global scope:

but non-virtual destructor

/home/ped/upp/uppsrc/Core/Value.h:374: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h:375: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h:376: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:380: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h:381: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:385: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:389: warning: comparison between signed and unsigned integer expressions

/home/ped/upp/uppsrc/Core/Value.h:393: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h: At global scope:

but non-virtual destructor

/home/ped/upp/uppsrc/Core/Value.h:554: warning: comparison between signed and unsigned integer expressions
/home/ped/upp/uppsrc/Core/Value.h: At global scope:

non-virtual destructor

non-virtual destructor

functions but non-virtual destructor

St>::Expand() [with T = char, S = Upp::Stri

St>::Cat(int) [with T = char, S = Upp::S

/home/ped/upp/uppsrc/Core/String.h:321:   instantiated from here

St>::Expand(int) [with T = char, S = Upp::S

St>::Cat(int, int) [with T = char, S = U

/home/ped/upp/uppsrc/Core/String.h:322:   instantiated from here

St>::Expand() [with T = short unsigned int,

St>::Cat(int) [with T = short unsigned i

/home/ped/upp/uppsrc/Core/String.h:618:   instantiated from here

St>::Expand(int) [with T = short unsigned i

St>::Cat(int, int) [with T = short unsig

/home/ped/upp/uppsrc/Core/String.h:619:   instantiated from here

EDIT:
(that's just includes, compiling the Core package itself will yield even more of them)

I tried to fix some of them, except missing virtual destructors I have got everything fixed, and I think it will work as supposed.
Yet the things like

```
template <class T, class S>
void AStringBuffer<T, S>::Expand(int len)
{
 /*typename S::Data *d = */S::GetData(begin);
 Realloc(max(8, max((int)(intptr_t)(alloc - begin + len), 2 * (int)(intptr_t)(alloc - begin))));
}
```

do make me a bit uncomfortable. I don't understand what "typename S::Data *d" really means and if commenting it out is ok.