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.

Subject: Re: Compiling U++ Core with -Wall (GCC 4.1.2 Kubuntu 6.10)
Posted by mirek on Thu, 18 Oct 2007 20:29:29 GMT
View Forum Message <> Reply to Message

Well, but it is 2007.1, is not it?

(There is no AStringBuffer in current U++).

Mirek

Subject: Re: Compiling U++ Core with -Wall (GCC 4.1.2 Kubuntu 6.10)
Posted by mr_ped on Thu, 18 Oct 2007 20:40:32 GMT

View Forum Message <> Reply to Message

Of course... sorry, I can't work with development versions right now.

a) I need pretty much stable environment, so even when 2007.2 will be out, it will take some tests and evaluation before I will be able to deploy it into my current development environment.

b) the development versions are worked on in windows, thus making the linux versions something like by-product ... adding even more uncertainty and hitting the stability issue. Doesn't look trustworthy for me right now.

Actually I don't need any new features for my current project, but I need highly stable and bug free code base which will eventually get trough external reviews and certifications.

This is still more than 12-18 months away, but that's the reason why I'm already starting to look into it, like trying "-Wall", and some more things in near future.

So I would love to have the U++ code base a tad cleaner in 2007.2 or .3 and next versions.

Subject: Re: Compiling U++ Core with -Wall (GCC 4.1.2 Kubuntu 6.10)
Posted by Novo on Thu, 18 Oct 2007 20:50:53 GMT

View Forum Message <> Reply to Message

luzr wrote on Thu, 18 October 2007 16:29Well, but it is 2007.1, is not it?

(There is no AStringBuffer in current U++).

Mirek

There are no src packages for two last development releases ...
And no info how to get source code elsewhere, except of digging it out of the windows release ...