Subject: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Novo on Wed, 01 Apr 2015 02:42:36 GMT
View Forum Message <> Reply to Message

I'm using TDM-GCC-64 (64-bit compiler on 64-bit Windows) and trying to compile a 32-bit version of my app.
I'm using the -m32 option with both compiler and linker.
The problem is with the resource compiler. It compiles into 64-bit resources.

windres.exe takes a --target option, which can be one from the list: pe-x86-64 pei-x86-64 pe-bigobj-x86-64 elf64-x86-64 elf64-l1om elf64-k1om pe-i386 pei-i386 elf32-i386 elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex.

There is no way to pass an option to the resource compiler, and there is only one GCC builder (I believe, TheIDE use to have both 32 and 64 bit builders for GCC).

Is there a way to fix that? I really need MINGW because I need GCC symbol demangling.

TIA

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by koldo on Wed, 01 Apr 2015 10:13:52 GMT
View Forum Message <> Reply to Message

Hello Novo

There are TDM-GCC 32 and 64 bits. You can install both in different folders (MINGW32 and MINGW64 by default).

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Novo on Wed, 01 Apr 2015 13:41:59 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 01 April 2015 06:13Hello Novo

There are TDM-GCC 32 and 64 bits. You can install both in different folders (MINGW32 and MINGW64 by default).

Hello koldo,

Thanks! Yes, I know, but TDM-GCC-64 is capable of compiling both 32 and 64 bit apps.
As a temporary solution I can install 32-bit version, but in general case it is not enough to have just a GCC builder. I can pass extra-arguments to compiler and linker, but windres.exe.

Thanks.

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by mirek on Wed, 01 Apr 2015 15:59:18 GMT
View Forum Message <> Reply to Message

Novo wrote on Wed, 01 April 2015 04:42I'm using TDM-GCC-64 (64-bit compiler on 64-bit Windows) and trying to compile a 32-bit version of my app.
I'm using the -m32 option with both compiler and linker.
The problem is with the resource compiler. It compiles into 64-bit resources.

windres.exe takes a --target option, which can be one from the list: pe-x86-64 pei-x86-64 pe-bigobj-x86-64 elf64-x86-64 elf64-l1om elf64-k1om pe-i386 pei-i386 elf32-i386 elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex.

There is no way to pass an option to the resource compiler, and there is only one GCC builder (I believe, TheIDE use to have both 32 and 64 bit builders for GCC).

Is there a way to fix that? I really need MINGW because I need GCC symbol demangling.

TIA

Hi,

thanks for the info.

I guess the quick fix and sufficient is to change the build method. I can do that, but it should be relatively easy to do (and I do not want to download/install mingw-tdm right now). If you succeed, please let me know, I will patch trunk. If you have problems, please report as well - I will try to fix it myself ASAP.

On related note, how is mingw-tdm working for you? Which version have you downloaded? I am (for some time now) interested in bundling it with U++ once again....

Mirek

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Novo on Sat, 19 Dec 2015 04:28:14 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 01 April 2015 11:59Novo wrote on Wed, 01 April 2015 04:42I'm using TDM-GCC-64 (64-bit compiler on 64-bit Windows) and trying to compile a 32-bit version of my app.
I'm using the -m32 option with both compiler and linker.
The problem is with the resource compiler. It compiles into 64-bit resources.

windres.exe takes a --target option, which can be one from the list: pe-x86-64 pei-x86-64 pe-bigobj-x86-64 elf64-x86-64 elf64-l1om elf64-k1om pe-i386 pei-i386 elf32-i386 elf64-little elf64-big elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex.

There is no way to pass an option to the resource compiler, and there is only one GCC builder (I believe, TheIDE use to have both 32 and 64 bit builders for GCC).

Is there a way to fix that? I really need MINGW because I need GCC symbol demangling.

TIA

Hi,

thanks for the info.

I guess the quick fix and sufficient is to change the build method. I can do that, but it should be relatively easy to do (and I do not want to download/install mingw-tdm right now). If you succeed, please let me know, I will patch trunk. If you have problems, please report as well - I will try to fix it myself ASAP.

On related note, how is mingw-tdm working for you? Which version have you downloaded? I am (for some time now) interested in bundling it with U++ once again....

Mirek

Hi Mirek,

Sorry for the delay with the answer. I just missed your message. E-mail notification doesn't work for me after the forum update.

I'm using mingw-tdm 4.9.2 for x64 target and 4.8.1 for x86. 5.1 is just too buggy, but you already know that at this time.

I still couldn't compile for x86 using 64-bit compiler. windres.exe is still not getting a --target option.

Thanks

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by omari on Sat, 19 Dec 2015 15:13:20 GMT
View Forum Message <> Reply to Message

patch:
file GccBuilder.icpp
line 226

```
exec << "windres -i " << GetHostPathQ(fn) <<  ((HasFlag("WIN32"))? " --target=pe-i386 " : "")
```

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by mirek on Mon, 18 Jan 2016 09:19:01 GMT
View Forum Message <> Reply to Message

Thanks. WIN32 is defined for WIN64 too, so the fix actually has to search the commandline for -m32 flag. No big deal. Seems to work.

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by pfsdanny on Fri, 09 Sep 2016 07:58:38 GMT
View Forum Message <> Reply to Message

I just upgrade U++ from 7962 to 9251 (64 bit), when comply the program I got the following error

C:\upp9251\bin/TDM64/bin/ld.exe: i386:x86-64 architecture of input file
`C:/Users/danny/Documents/uppout9251/MEC9251/Elink/MINGW.For
ce_Speed.Gui.Main.Mt.Mysqldll\resource$rc.o' is incompatible with i386 output

I try your patch but cannot fix the error. When I remove the recource.rc from the project, everything is fine. I need the resource.rc to embed the application icon in the exe.

Is there any fix?

Thanks.

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Melek on Mon, 23 Jan 2017 16:37:07 GMT
View Forum Message <> Reply to Message

Jejeje I have the same error and tried several ways.
The IDE of U ++ is really cool, however not being able to compile or use the resource file is complicated to move forward, and I add that there is almost no documentation on the web, although TheIde already comes with a lot of examples which Is really excellent, but if it would be nice to have more documentation and support with these errors, because I can not place the .ico file I can only compile with .iml file which only places the icon in the window and taskbar but not in the executable Which is really important. I also see that there is no option unless it is for an .rc file to place compile details for the executable, such as:
-Version
-Description
-Author
-Brand
-Etc

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW

Posted by mirek on Mon, 23 Jan 2017 23:23:18 GMT
View Forum Message <> Reply to Message

Can you please try with the lastest nightly build?

I have just checked current nigtly, MINGW (which is 32bit) with examples/EyeCare (which has .rc file) and it seems to work just fine...

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Melek on Tue, 24 Jan 2017 01:55:48 GMT
View Forum Message <> Reply to Message

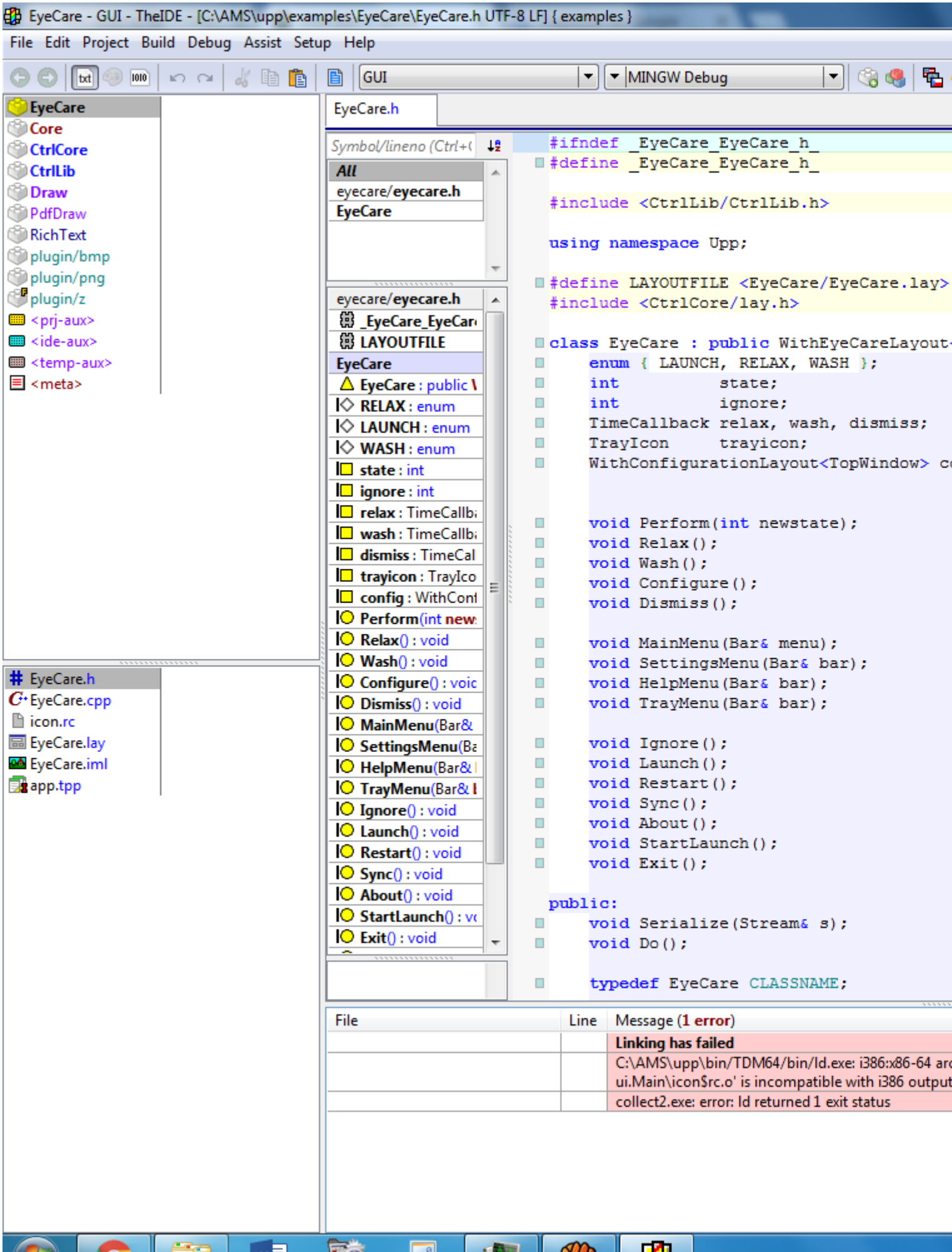
Thanks for the prompt response and take the opportunity to indicate that I really loved this framework, that's why I'm trying it, although I'm adapting.
I tried EyeCare and Bumps. But the problem is the same, the file I see is compiled into resuource $ rc.o but throws the error that is incompatible with i386. Here is the output:

C:\AMS\upp\bin/TDM64/bin/ld.exe: i386:x86-64 architecture of input file
`C:/AMS/upp/out/examples/EyeCare/MINGW.Debug.Debug_Full.Gui. Main\ic
on$rc.o' is incompatible with i386 output

collect2.exe: error: ld returned 1 exit status

File Attachments

1) Sin título.png , downloaded 527 times


EyeCare - GUI - TheIDE - [C:\AMS\upp\examples\EyeCare\EyeCare.h UTF-8 LF] { examples }
File Edit Project Build Debug Assist Setup Help
GUI
MINGW Debug
EyeCare
Core
CtrlCore
CtrlLib
Draw
PdfDraw
RichText
plugin/bmp
plugin/png
plugin/z
<prj-aux>
<ide-aux>
<temp-aux>
<meta>
EyeCare.h
EyeCare.cpp
icon.rc
EyeCare.lay
EyeCare.iml
app.tpp
EyeCare.h
Symbol/lineno (Ctrl+(
All
eyecare/eyecare.h
EyeCare
eyecare/eyecare.h
_EyeCare_EyeCar
LAYOUTFILE
EyeCare
EyeCare : public \
RELAX : enum
LAUNCH : enum
WASH : enum
state : int
ignore : int
relax : TimeCallb
wash : TimeCallb
dismiss : TimeCal
trayicon : TrayIco
config : WithConf
Perform(int new
Relax() : void
Wash() : void
Configure() : voic
Dismiss() : void
MainMenu(Bar&
SettingsMenu(Ba
HelpMenu(Bar&
TrayMenu(Bar&
Ignore() : void
Launch() : void
Restart() : void
Sync() : void
About() : void
StartLaunch() : v
Exit() : void
#ifndef _EyeCare_EyeCare_h_
#define _EyeCare_EyeCare_h_
#include <CtrlLib/CtrlLib.h>
using namespace Upp;
#define LAYOUTFILE <EyeCare/EyeCare.lay>
#include <CtrlCore/lay.h>
class EyeCare : public WithEyeCareLayout
enum { LAUNCH, RELAX, WASH };
int state;
int ignore;
TimeCallback relax, wash, dismiss;
TrayIcon trayicon;
WithConfigurationLayout<TopWindow> c
void Perform(int newstate);
void Relax();
void Wash();
void Configure();
void Dismiss();
void MainMenu(Bar& menu);
void SettingsMenu(Bar& bar);
void HelpMenu(Bar& bar);
void TrayMenu(Bar& bar);
void Ignore();
void Launch();
void Restart();
void Sync();
void About();
void StartLaunch();
void Exit();
public:
void Serialize(Stream& s);
void Do();
typedef EyeCare CLASSNAME;
File
Line
Message (1 error)
Linking has failed
C:\AMS\upp\bin/TDM64/bin/ld.exe: i386:x86-64 ar
ui.Main\icon$rc.o' is incompatible with i386 output
collect2.exe: error: ld returned 1 exit status

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Melek on Tue, 24 Jan 2017 01:59:44 GMT
View Forum Message <> Reply to Message

Latest nightly build is beta or can it be used without problems?

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Melek on Tue, 24 Jan 2017 03:13:33 GMT
View Forum Message <> Reply to Message

Excellent, I downloaded the lastest nightly build and it worked perfectly Thank you very much

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by mirek on Tue, 24 Jan 2017 07:10:59 GMT
View Forum Message <> Reply to Message

Melek wrote on Tue, 24 January 2017 02:59Latest nightly build is beta or can it be used without problems?

Except glitches, latest nightly tends to be more stable than official release.

Right now it is 99.99% stable as we are close to 2017.1 release.

Mirek

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Melek on Tue, 24 Jan 2017 21:28:07 GMT
View Forum Message <> Reply to Message

Excelent Good Job.
I already tested with "2017.1rc1" and it worked perfectly for me to place the icon and version details in the .rc file
I imagine that using the MinGW compiler the executable is compiled in machine language.
And I was seeing some tutorials on the page and I would like to know if a window created as file.lay can be made activate MaximizeBox and MinimizeBox, since that can only do it from the code directly. And I still do not quite understand how to create several window2.cpp, window3.cpp classes as dialog type windows using layout assistance "window2.lay and window3.lay". These are some details that I do not know yet. I have tried with the examples that appear on the page to create modal windows and it works, but suddenly for large applications, it would be easier to create windows in their own classes and files to make code maintenance more comfortable.

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW

Posted by mirek on Wed, 25 Jan 2017 07:57:44 GMT
View Forum Message <> Reply to Message

Melek wrote on Tue, 24 January 2017 22:28 Excelent Good Job.
I already tested with "2017.1rc1" and it worked perfectly for me to place the icon and version details in the .rc file
I imagine that using the MinGW compiler the executable is compiled in machine language.
And I was seeing some tutorials on the page and I would like to know if a window created as file.lay can be made activate MaximizeBox and MinimizeBox,

Activate by TopWindow Sizeable and Zoomable methods (e.g. put into dialog constructor).

Quote:
since that can only do it from the code directly. And I still do not quite understand how to create several window2.cpp, window3.cpp classes as dialog type windows using layout assistance "window2.lay and window3.lay". These are some details that I do not know yet. I have tried with the examples that appear on the page to create modal windows and it works, but suddenly for large applications, it would be easier to create windows in their own classes and files to make code maintenance more comfortable.

Sure, that is how it is done.

General note: There is one really big 'example' of U++: TheIDE sources... (just load uppsrc/ide project)

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Melek on Wed, 25 Jan 2017 21:16:39 GMT
View Forum Message <> Reply to Message

Excellent I'm reviewing, it's something complicated to understand perfectly because I'm starting to know the language, although the great amount of examples they have is extraordinary and that has helped me a lot.
I already handle other languages interpreted as Java and PHP.
But C ++ I like because the compiled files are created in machine language which supposes a security mayo as opposed to .Net and Java.
I will be checking everything and thank you again

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by Melek on Thu, 26 Jan 2017 21:12:49 GMT
View Forum Message <> Reply to Message

New Topic. Plz Help me
http://www.ultimatepp.org/forums/index.php?t=msg&th=9875 &start=0&

Subject: Re: Problem compiling 32-bit apps with 64-bit MINGW
Posted by mr_ped on Fri, 27 Jan 2017 01:05:35 GMT
View Forum Message <> Reply to Message

Melek wrote on Wed, 25 January 2017 22:16
But C ++ I like because the compiled files are created in machine language which supposes a security mayo as opposed to .Net and Java.

I'm not sure what is "mayo", but unless you have group of seasoned C++ developers with years of experience, it's pretty sure your C++ application will be much worse than Java/C# in terms of security (ie. much easier to exploit and containing many security vulnerabilities).

If you have several C++ senior developers and good development process in place, like security reviews, using valgrind and similar tools, penetration testing, etc... then you can produce secure software even with C++, but you have to spend considerable amount of budget on the security. Then again this true also for Java/C# at this level.

But if you just create some small app in few hours, then it's more likely the Java/C# will be somewhat secure even without paying attention to it, while C++ goes the opposite way, without paying the attention to it it will be almost surely insecure.