MinGW has updated the MinGW C++ compiler in late December 2018 from version 6.3.0 to version 8.2.0 of the C++ compiler. This is great, but there is a problem with the compiler recognizing the

-std=c++11, -std=c++14 and -std=c++17 compiler switches as it did in previous versions.

By default version 8.2.0 of the MinGW C++ compiler will use the C++14 standard. So, in order to get the compiler working, you should NOT check any the C++ compiler switches that change the version of C++ used by the compiler. Please see the image below.

TheProblem Project Setting	IS	$\times$
Debug		~
General Compiler Linker Environment Debugger Resources Pre / Post Build Comman Customize Code Completion Global Settings QMake	Compiler   C++ Compiler Options   C Compiler Options   Additional Include Paths   Compiler Options   Additional Include Paths   Compiler Options   Check the command line options you need   @Enable all compiler warnings [-Wall]   @Enable ANSI C99 features [-std=c99]   @Enable C++11 features [-std=c++11]   @Enable C++11 features [-std=c++11]   @Enable C++11 features [-std=c++17]   @Enable Standard compiler warnings [-W]   @Enable warnings demanded by strict ISO C and ISO C++ [-pedantic]   @Expensive optimizations [-fexpensive-optimizations]   [In C mode, support all ISO C90 programs. In C++ mode, remove GNU extensions that conflict with ,   @Enter other options	
	Command line -Wall	
	OK Cancel	

If you wish to use another C++ standard besides the C++14 standard, you can use the following compiler switches:

-std=gnu++11 -std=gnu++14 -std=gnu++17

You can enter these switches manually in the project settings (and then save the changes to another template) or you can modify the switch commands generated by CodeLite as follows:

Select Settings->Build Settings

Then select the MinGW compiler from the left side and Compiler Options from the right side. See the image below:

Compilers Build Ou	tput Appearance Build Systems		
+ ti ini			
/linGW ( MinGW )	Tools Patterns Compiler Options Linker Options Switches File Types Advance	d	
	Double click on an entry to modify it:		
	Switch Help	^	New
	-O Optimize generated code. (for speed)		
	-O0 Optimize for debugging		Delete
	-O1 Optimize more (for speed)		
	-O2 Optimize even more (for speed)		
	-O3 Optimize fully (for speed)		
	-Os Optimize generated code (for size)		
	-W Enable standard compiler warnings		
	-Wall Enable all compiler warnings		
	-Wfat Stop compiling after first error		
	-Wm Warn if main() is not conformant		
	-ansi In C mode, support all ISO C90 programs. In C++ mode, remove GNU extensions	th	
	-fexp Expensive optimizations	~	
	٢	>	

Then select the compiler switch you want to modify and change it. For example, for C++17 you would change the switch from –std=c++17 to –std=gnu++17. See the image below:

Build Settings									
Compilers Build Out	put Appeara	nce Buil	d Systems						
+ 1 🖻									
MinGW ( MinGW )	Tools	Patterns	Compiler Options	Linker Option	s Switches	File Types	Advanced		
	Double cl	ick on an e	entry to modify it:						
	Switch	Help						New	
	-ansi	n C mode,	support all ISO C90	programs. In C	++ mode, re	move GNU e	xtensions the		
			optimizations Co	Compiler option					
			Enable OpenMP (compilatio						
				Jgging informa Switch: -std=gnu++17					
			rnings demanded rors the warnings Help: Enable C++17 features						
			e when executed	ings -					
		Enable C+-	+11 features						
	-std=	Enable C+-	+14 features						
	-std=	Enable C+-	+17 features						
			SI C99 features		01/	Cana	-1		
		nhibit all v	varning message		OK	Canc	el		
	<						>		
			ОК	Cancel	Apply				

Hopefully the MinGW developers will again support the standard switches in a future release.

If you are using the command-line to compiler, please provide the updated compiler switch as you normally would.

Best regards, Frank