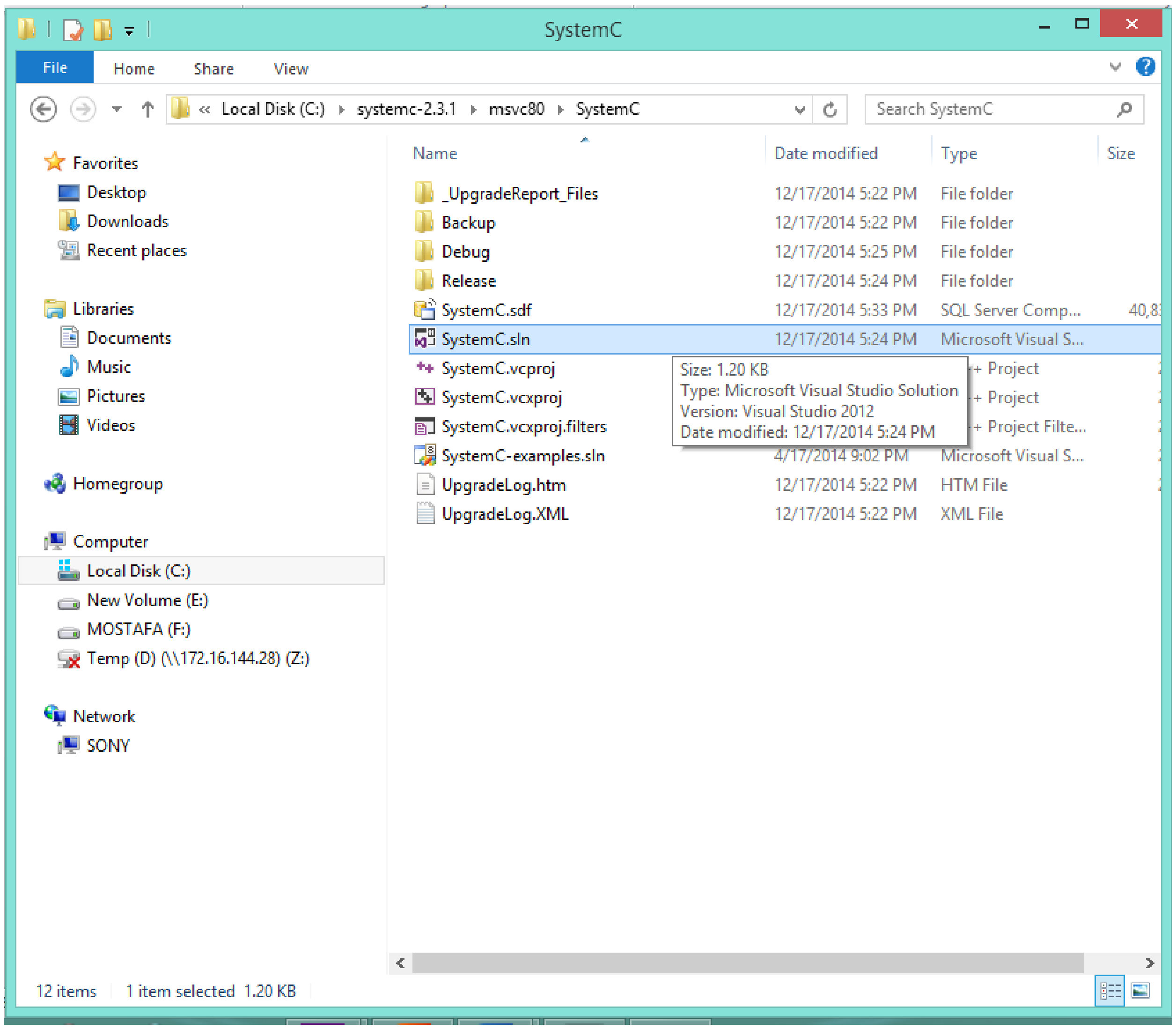
Installation systemc-2.3.1 on VS2013

Download systemc-2.3.1 from http://www.accellera.org/downloads or http://s5.picofile.com/d/e37381b3-7f5d-403c-aec8-dac03cd157eb/systemc_2_3_1.tgz

Unzip this file and put them in c: (just internal systemc-2.3.1 file).

Double click on C:\systemc-2.3.1\msvc80\SystemC\ SystemC.sln

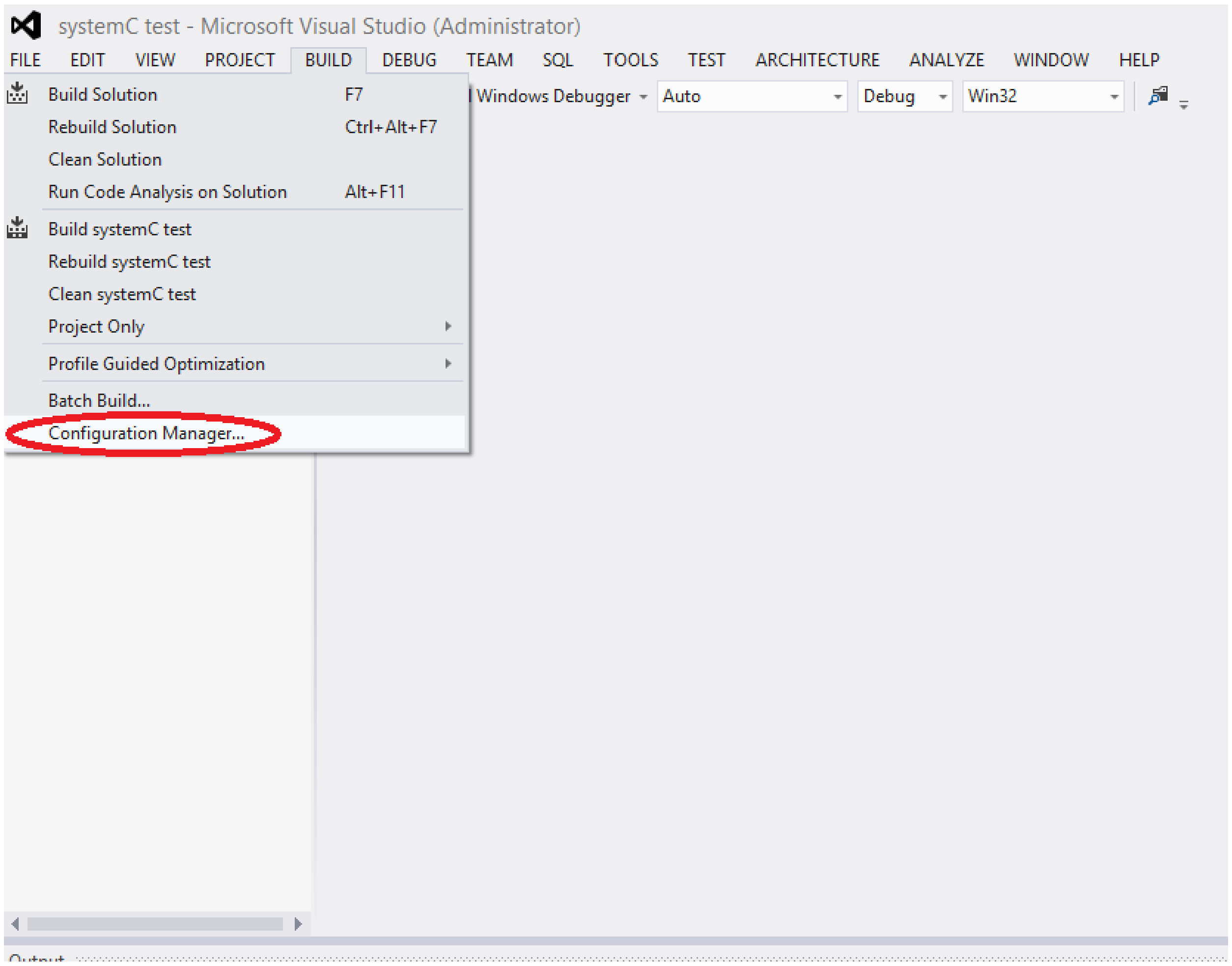


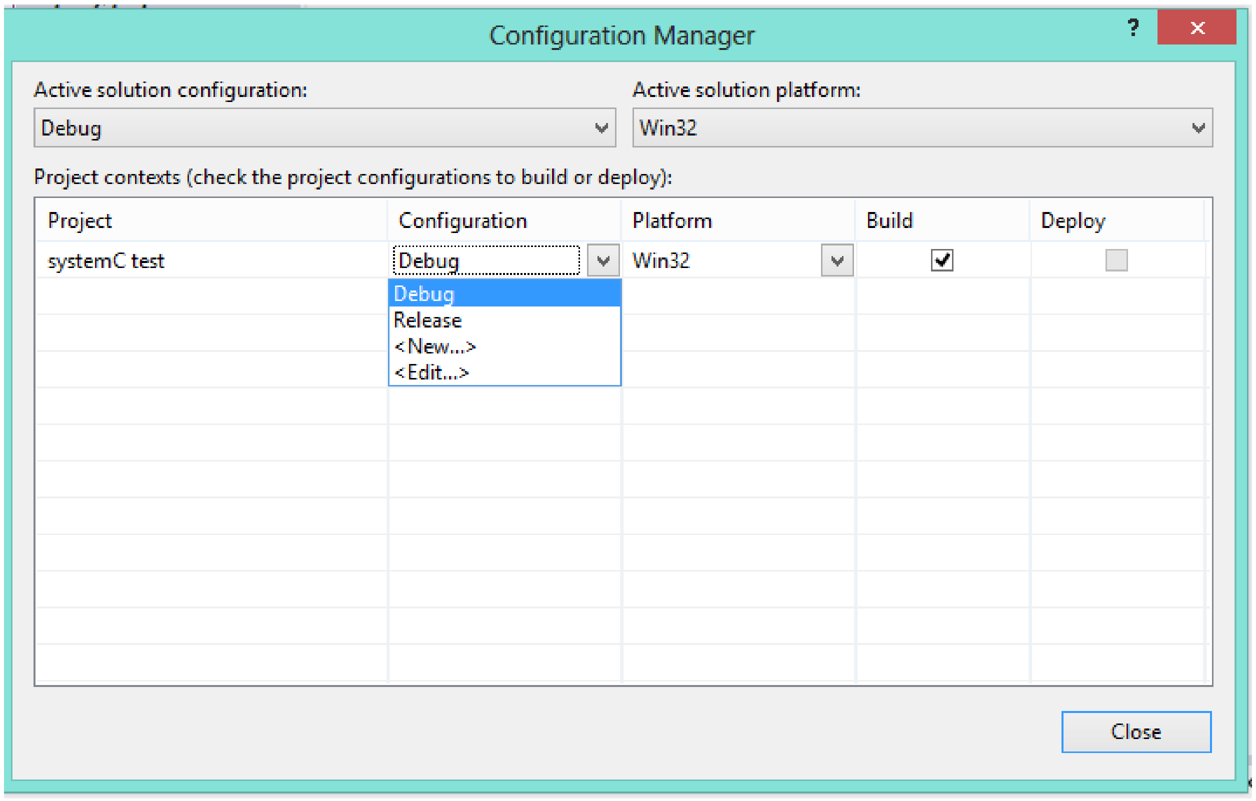
Name	Date modified	Type	Size
_UpgradeReport_Files	12/17/2014 5:22 PM	File folder	
Backup	12/17/2014 5:22 PM	File folder	
Debug	12/17/2014 5:25 PM	File folder	
Release	12/17/2014 5:24 PM	File folder	
SystemC.sdf	12/17/2014 5:33 PM	SQL Server Comp...	40,8
SystemC.sln	12/17/2014 5:24 PM	Microsoft Visual S...	
SystemC.vcproj		+ Project	
SystemC.vcxproj		+ Project	
SystemC.vcxproj.filters		+ Project Filte...	
SystemC-examples.sln	4/17/2014 9:02 PM	Microsoft Visual S...	
UpgradeLog.htm	12/17/2014 5:22 PM	HTM File	
UpgradeLog.XML	12/17/2014 5:22 PM	XML File	

Size: 1.20 KB
Type: Microsoft Visual Studio Solution
Version: Visual Studio 2012
Date modified: 12/17/2014 5:24 PM

Then open vs13

Now, build Release and Debug:


systemC test - Microsoft Visual Studio (Administrator)
FILE
EDIT
VIEW
PROJECT
BUILD
DEBUG
TEAM
SQL
TOOLS
TEST
ARCHITECTURE
ANALYZE
WINDOW
HELP
Build Solution F7
Rebuild Solution Ctrl+Alt+F7
Clean Solution
Run Code Analysis on Solution Alt+F11
Build systemC test
Rebuild systemC test
Clean systemC test
Project Only
Profile Guided Optimization
Batch Build...
Configuration Manager...
Windows Debugger
Auto
Debug
Win32
Output


Configuration Manager

Active solution configuration: Debug

Active solution platform: Win32

Project contexts (check the project configurations to build or deploy):

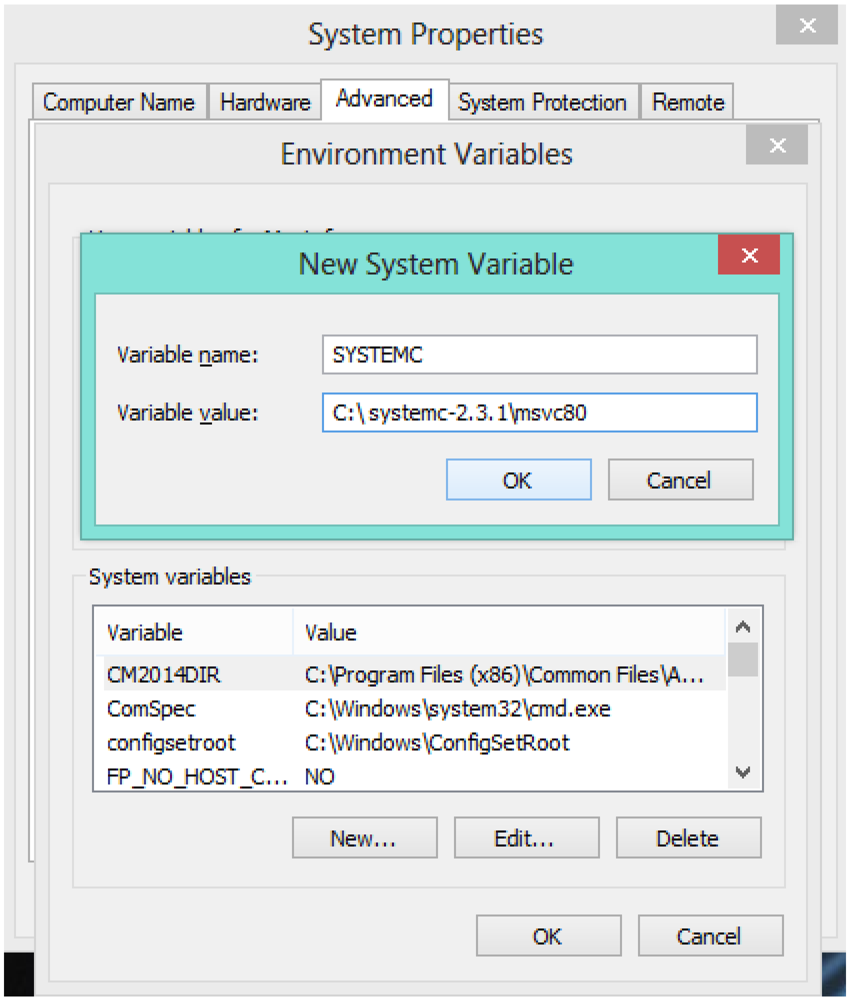
Project	Configuration	Platform	Build	Deploy
systemC test	Debug	Win32	✓	☐

Repeat for Debug/Release configuration. Set the Configuration back to Debug.
Make sure dates of Debug and Release folders have changed
C:\ systemc-2.3.1\msvc80\SystemC\Release\SystemC.lib
C:\ systemc-2.3.1\msvc80\SystemC\Debug\SystemC.lib

****

Now set system attributes (Windows Environment Variables):
Right-click on Computer

Go to Click on Advanced system setting and add SYSTEMC in system variable.



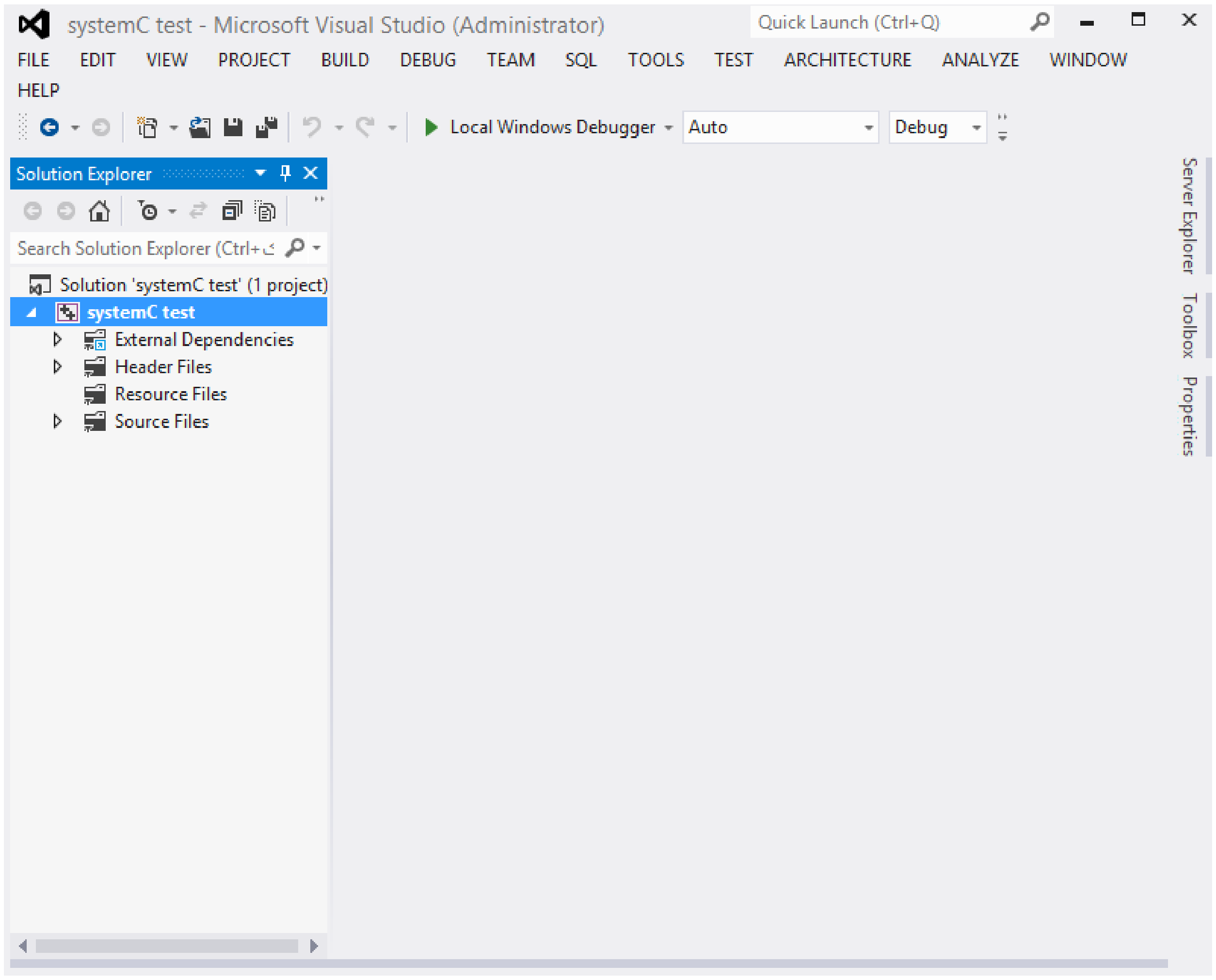
Variable name: SYSTEMC

Variable value: C:\ systemc-2.3.1\msvc80

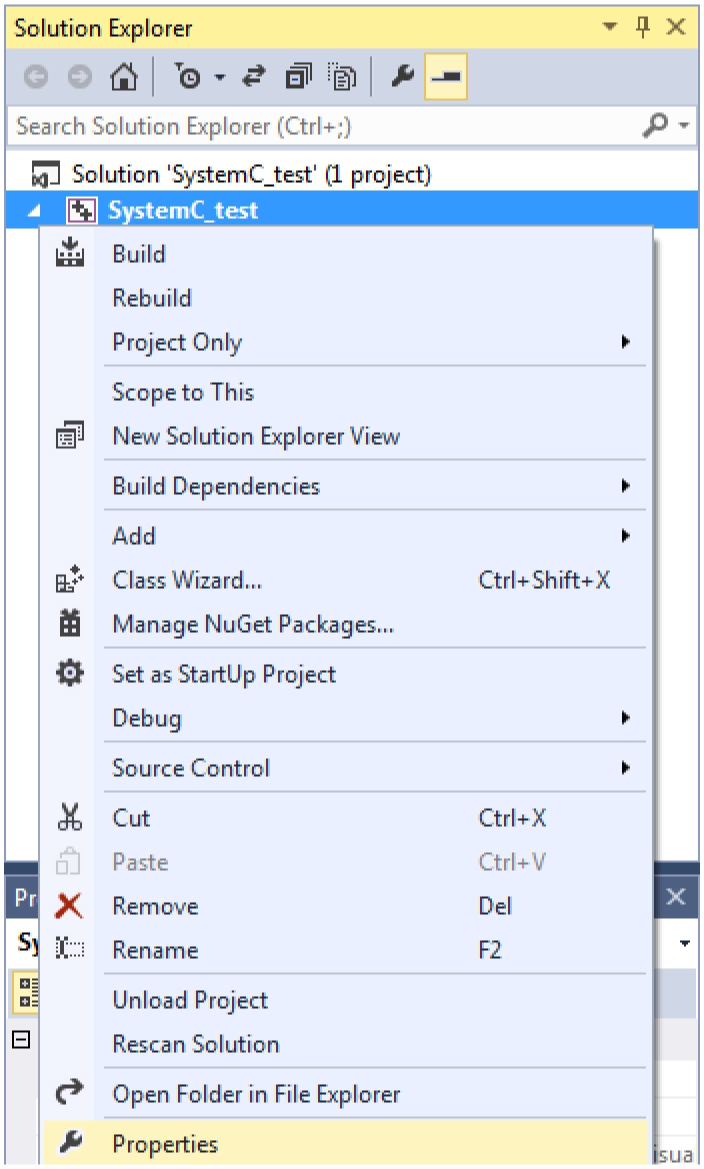
Variable	Value
CM2014DIR	C:\Program Files (x86)\Common Files\A...
ComSpec	C:\Windows\system32\cmd.exe
configsetroot	C:\Windows\ConfigSetRoot
FP_NO_HOST_C...	NO

Start VS13:

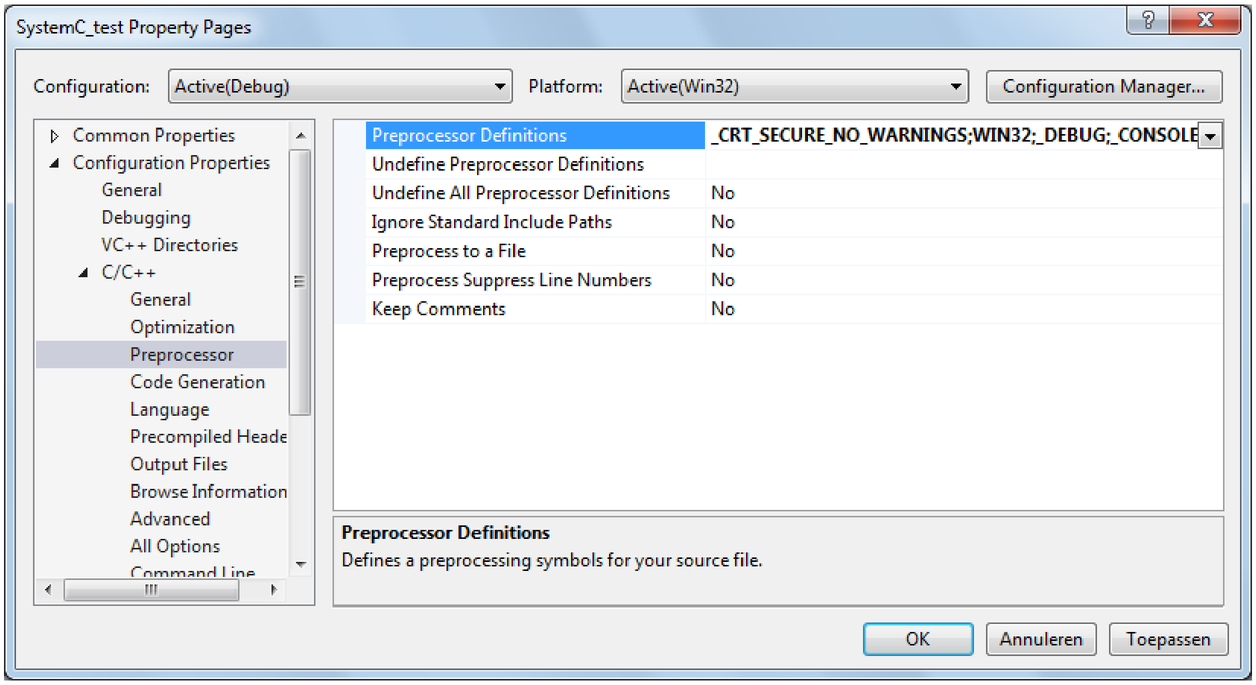
New Project
Win32 Console Application


systemC test - Microsoft Visual Studio (Administrator)
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Local Windows Debugger
Auto
Debug
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'systemC test' (1 project)
systemC test
External Dependencies
Header Files
Resource Files
Source Files
Server Explorer
Toolbox
Properties

Go to properties



Configuration Properties → C/C++ → Preprocessor: add _CRT_SECURE_NO_DEPRECATE;_CRT_SECURE_NO_WARNINGS; to definitions:

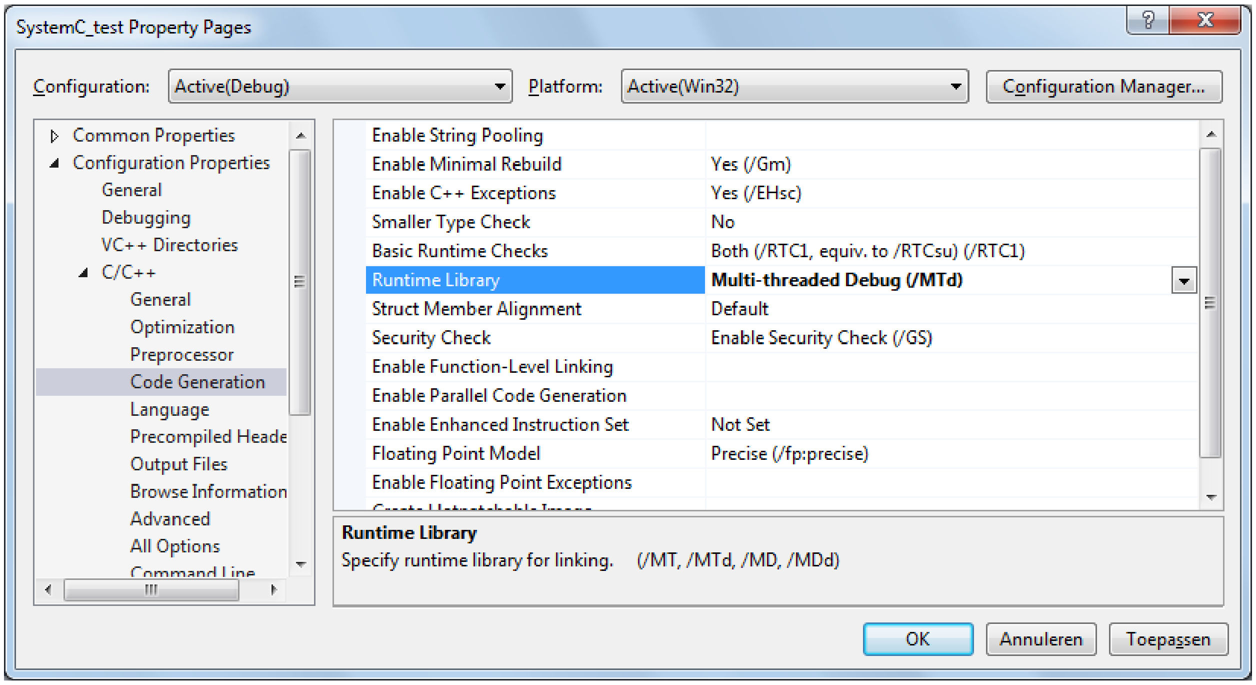

SystemC_test Property Pages

Configuration: Active(Debug) Platform: Active(Win32)

Preprocessor Definitions	_CRT_SECURE_NO_WARNINGS;WIN32;_DEBUG;_CONSOLE
Undefine Preprocessor Definitions	
Undefine All Preprocessor Definitions	No
Ignore Standard Include Paths	No
Preprocess to a File	No
Preprocess Suppress Line Numbers	No
Keep Comments	No

Preprocessor Definitions
Defines a preprocessing symbols for your source file.

Configuration Properties → C/C++ → Code Generation properties, set Runtime Library to Multi-threaded Debug (/MTd) for debug build, and /MT for release build:

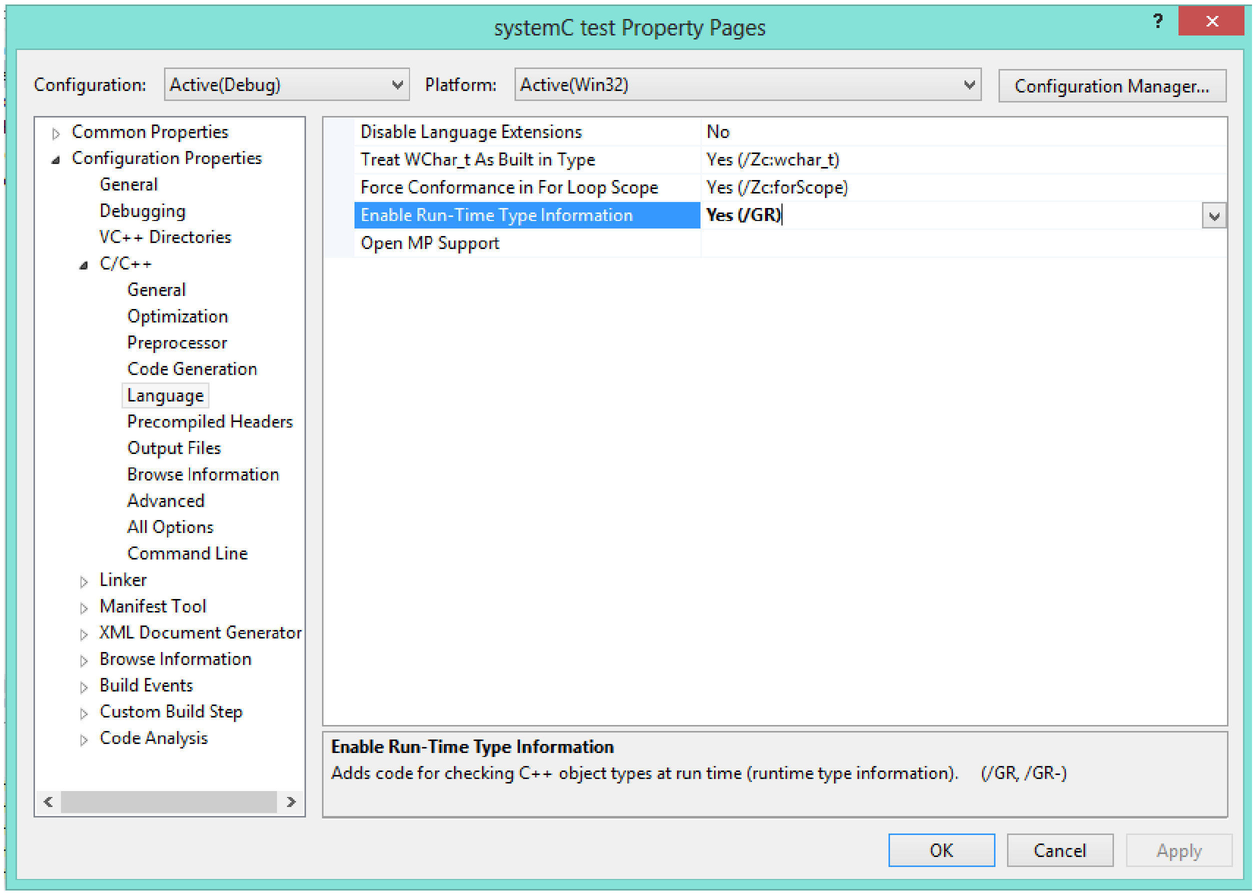

SystemC_test Property Pages

Configuration: Active(Debug) Platform: Active(Win32)

Enable String Pooling	
Enable Minimal Rebuild	Yes (/Gm)
Enable C++ Exceptions	Yes (/EHsc)
Smaller Type Check	No
Basic Runtime Checks	Both (/RTC1, equiv. to /RTCsu) (/RTC1)
Runtime Library	**Multi-threaded Debug (/MTd)**
Struct Member Alignment	Default
Security Check	Enable Security Check (/GS)
Enable Function-Level Linking	
Enable Parallel Code Generation	
Enable Enhanced Instruction Set	Not Set
Floating Point Model	Precise (/fp:precise)
Enable Floating Point Exceptions	

Runtime Library
Specify runtime library for linking. (/MT, /MTd, /MD, /MDd)

Configuration Properties → C/C++ → Language, set Enable Run-Time Type Info to Yes:



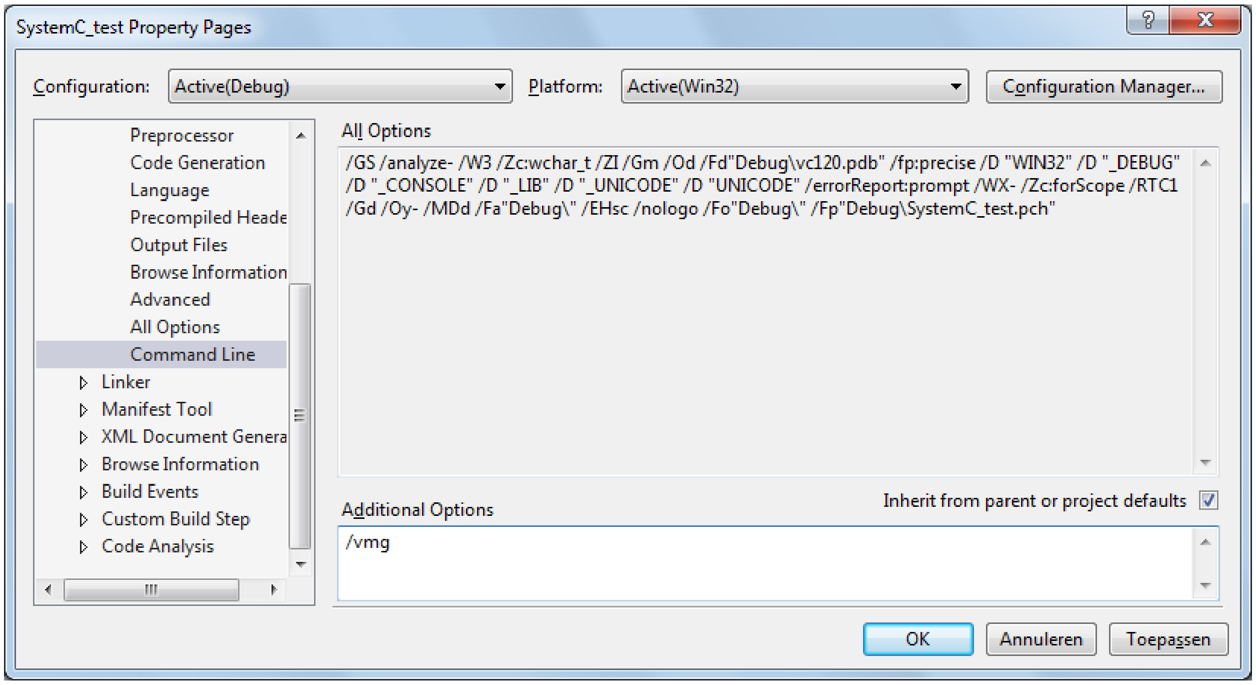
systemC test Property Pages

Configuration: Active(Debug)   Platform: Active(Win32)

Disable Language Extensions	No
Treat WChar_t As Built in Type	Yes (/Zc:wchar_t)
Force Conformance in For Loop Scope	Yes (/Zc:forScope)
Enable Run-Time Type Information	Yes (/GR)
Open MP Support	

Enable Run-Time Type Information
Adds code for checking C++ object types at run time (runtime type information).   (/GR, /GR-)

Configuration Properties → C/C++ → Command Line, add /vmg to Additional Options:



All Options

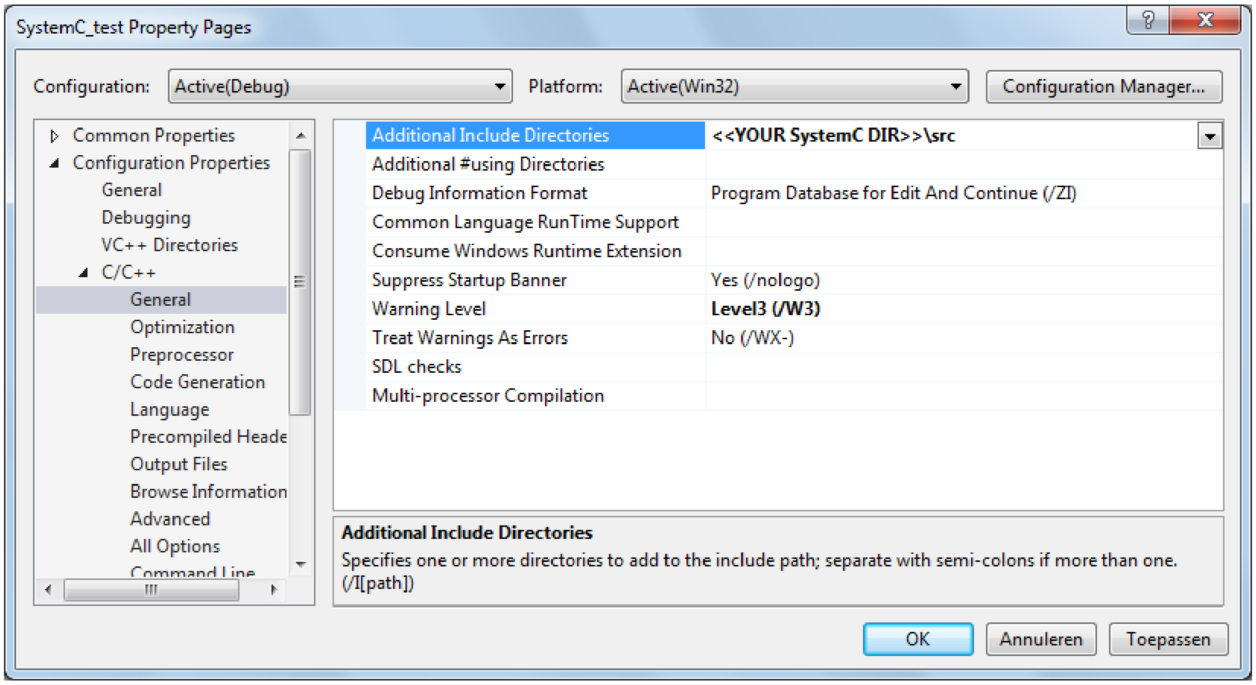
```
/GS /analyze- /W3 /Zc:wchar_t /ZI /Gm /Od /Fd"Debug\vc120.pdb" /fp:precise /D "WIN32" /D "_DEBUG"
/D "_CONSOLE" /D "_LIB" /D "_UNICODE" /D "UNICODE" /errorReport:prompt /WX- /Zc:forScope /RTC1
/Gd /Oy- /MDd /Fa"Debug\" /EHsc /nologo /Fo"Debug\" /Fp"Debug\SystemC_test.pch"
```

Additional Options

Inherit from parent or project defaults ✓

```
/vmg
```

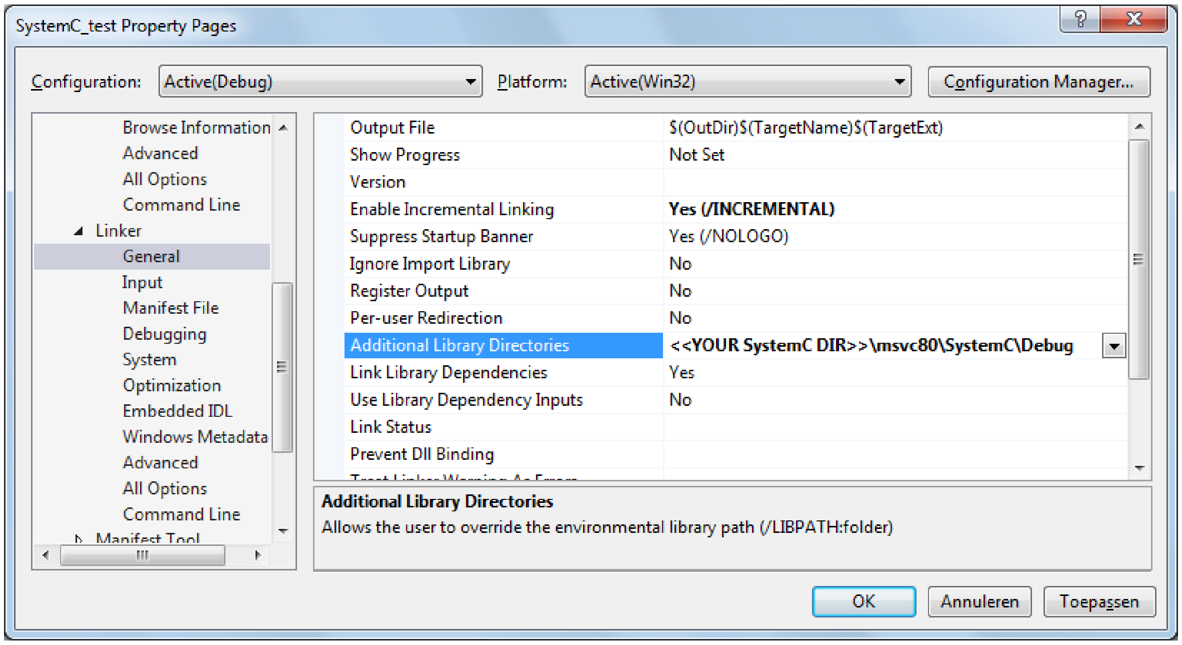
Configuration Properties → C/C++ → General, add [C:\systemc-2.3.1\src;%(AdditionalIncludeDirectories)] to Additional Include Directories:



Additional Include Directories	<<YOUR SystemC DIR>>\src
Additional #using Directories	
Debug Information Format	Program Database for Edit And Continue (/ZI)
Common Language RunTime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
SDL checks	
Multi-processor Compilation	

Additional Include Directories
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

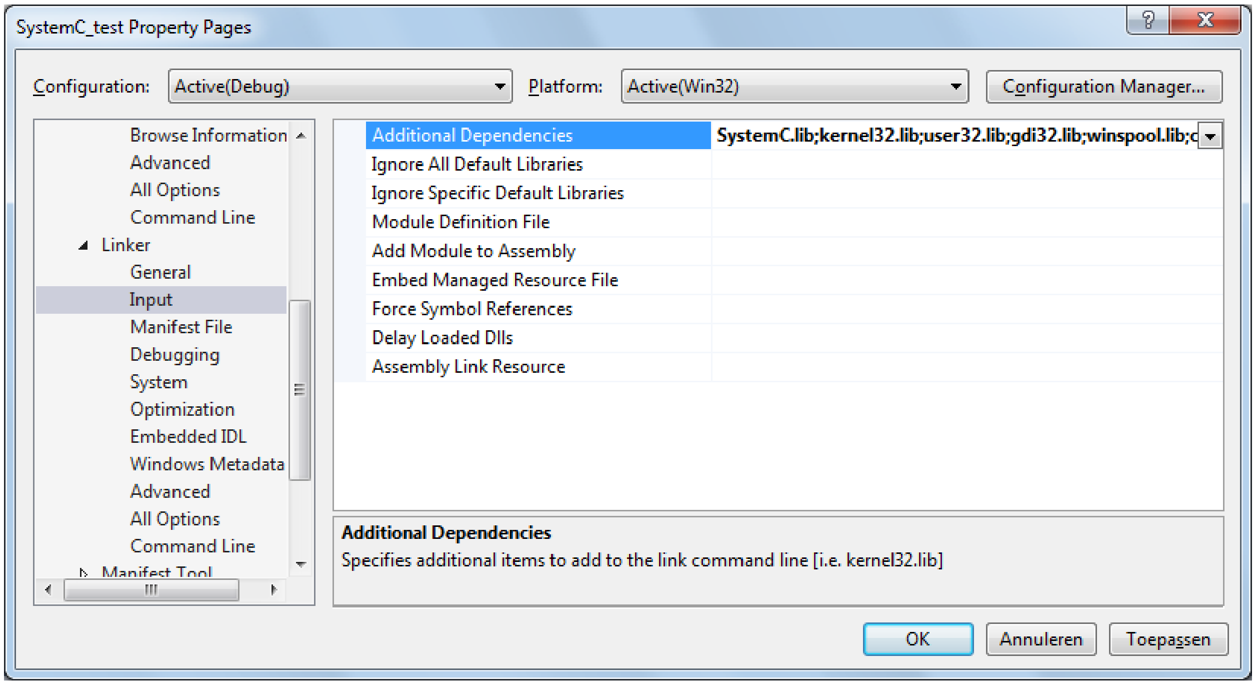
Configuration Properties → Linker → General, add C:\systemc-2.3.1\msvc80\SystemC\Debug;%(AdditionalLibraryDirectories) to Additional Library Directories.



Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	**Yes (/INCREMENTAL)**
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**<<YOUR SystemC DIR>>\msvc80\SystemC\Debug**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	

Additional Library Directories
Allows the user to override the environmental library path (/LIBPATH:folder)

Configuration Properties → Linker → Input, add SystemC.lib; in the Additional Dependencies:



Additional Dependencies	**SystemC.lib;kernel32.lib;user32.lib;gdi32.lib;winspool.lib;c**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line [i.e. kernel32.lib]

Click OK and run below example:

- `#include <systemc>`
-
- `using namespace sc_core;`
- `using namespace sc_dt;`

```
using namespace std;

/* Simple DFF */

SC_MODULE(dff) { /* Model of a Data Flip-Flop that reacts on a negative edge of the clock signal clk */
    sc_in_clk clk;
    sc_in<sc_logic> din;
    sc_out<sc_logic> dout;
    SC_CTOR(dff) {
        SC_METHOD(on_clk_neg);
        sensitive << clk.neg(); /* Execute process on_clk_neg on every negative edge of the clock signal clk */
    }
private:
    void on_clk_neg() {
         /* Behavior of DFF */
        dout.write(din.read());
    }
};

SC_MODULE(tb_dff) { /* Test bench for the DFF */
    sc_clock clk;
    sc_signal<sc_logic> din;
    sc_signal<sc_logic> dout;
    SC_CTOR(tb_dff): clk("clk",10,SC_NS,0.5), DUT("dff") {
        /* Connect test bench with DFF which is the device under test (DUT)*/
        DUT.din(din);
        DUT.dout(dout);
        DUT.clk(clk);
        SC_THREAD(main);
    }
private:
    dff DUT;
    void main() {
         /* test script */
        din.write(SC_LOGIC_0);
        wait(31, SC_NS);
        din.write(SC_LOGIC_1);
        wait(42, SC_NS);
        din.write(SC_LOGIC_0);
    }
};

int sc_main(int argc, char* argv[]) {
    tb_dff TB("tb_dff");

     /* Trace (record) signals */
    sc_trace_file *tf(sc_create_vcd_trace_file("trace"));
    tf->set_time_unit(1, SC_NS);
    sc_trace(tf, TB.clk, "clk");
    sc_trace(tf, TB.din, "din");
    sc_trace(tf, TB.dout, "dout");

    sc_start(100, SC_NS);

    sc_close_vcd_trace_file(tf);
    cin.get();
    return 0;
}
```

Enjoy it.

Ms.fathi@ut.ac.ir