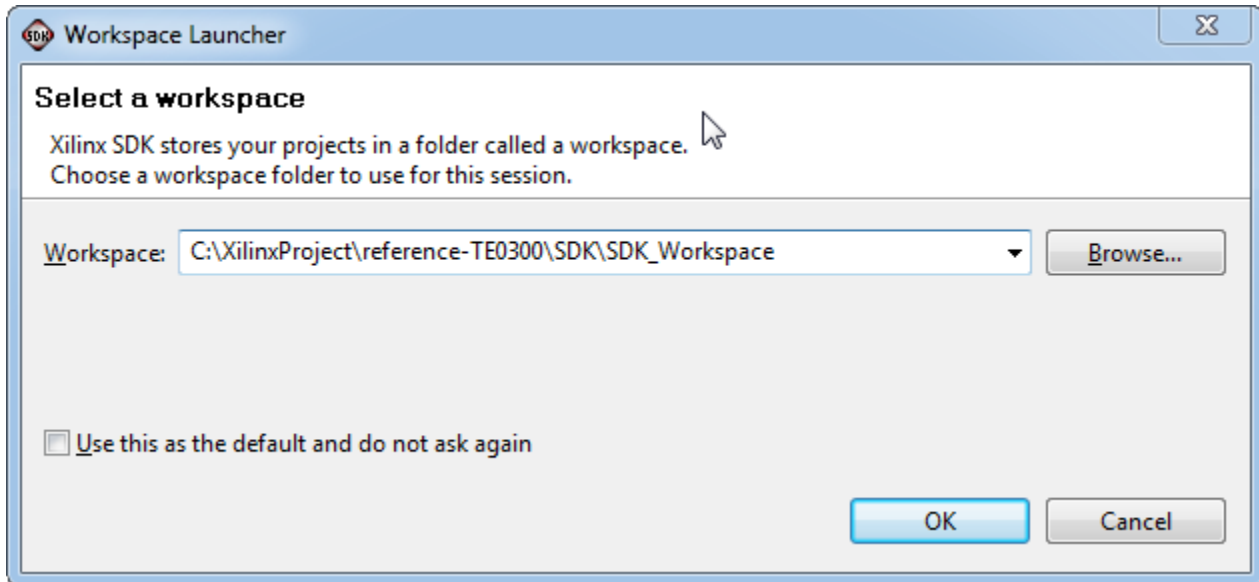


Update the SDK project

You should click "Start">"Xilinx Design Tools">"ISE Design Suite xx.x">"EDK">"Xilinx Software Development".

After this a pop-up will appear.



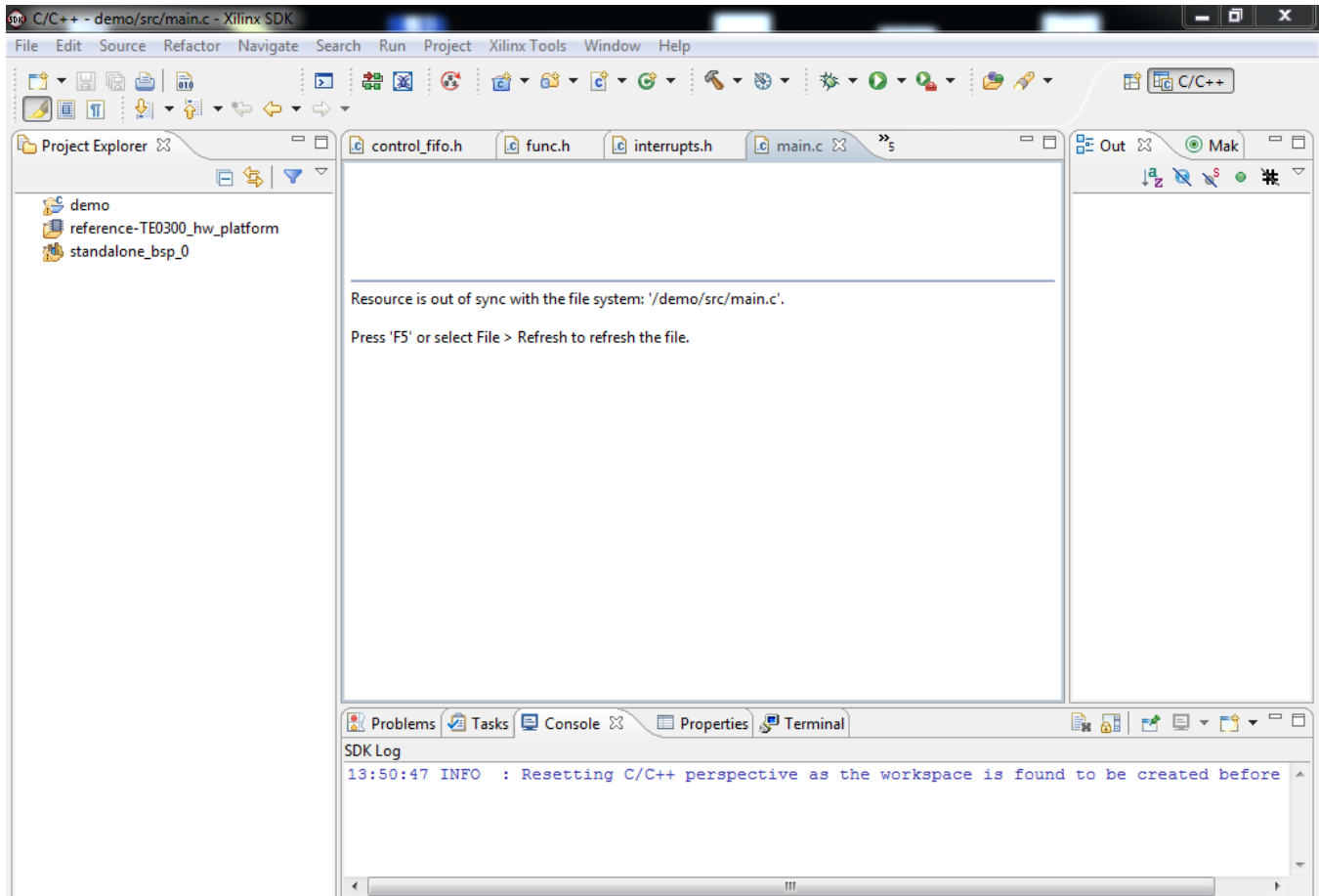
Select a workspace

In this pop-up you should choose the SDK_Workspace desired.



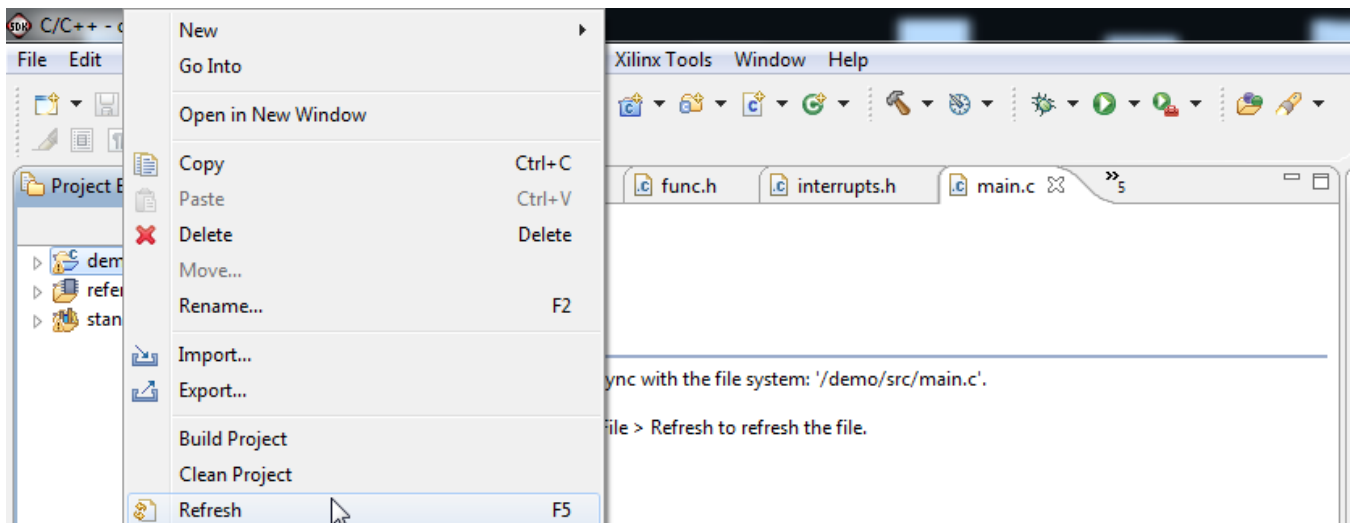
Workspace selected, SDK is loading

After this the SDK will start.



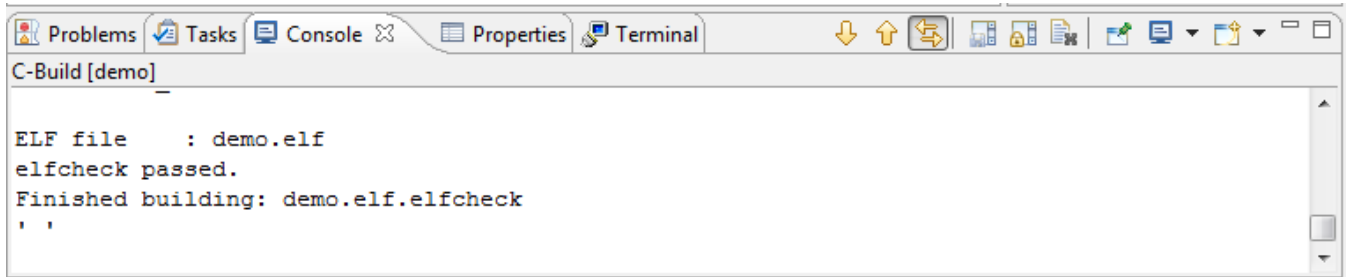
Xilinx SDK starts

If the version of the SDK is higher than the version of the project opened you should follow the instruction and refresh the demo .c and .h files.



Refresh .c and .h files of demo

After refresh operation this message should appear.



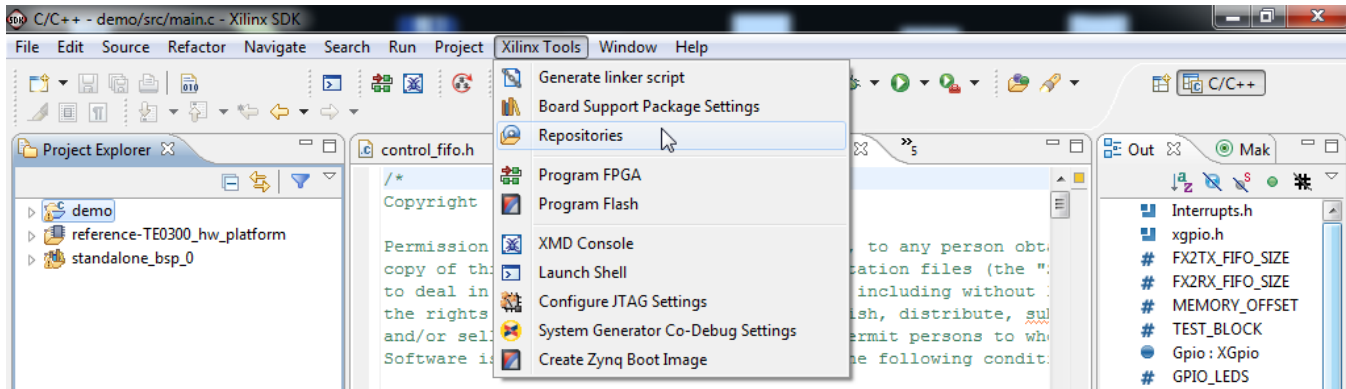
The screenshot shows the 'Console' tab of the Xilinx IDE. The title bar reads 'C-Build [demo]'. The output text is as follows:

```
ELF file      : demo.elf
elfcheck passed.
Finished building: demo.elf.elfcheck
' '
```

End of the refresh operation

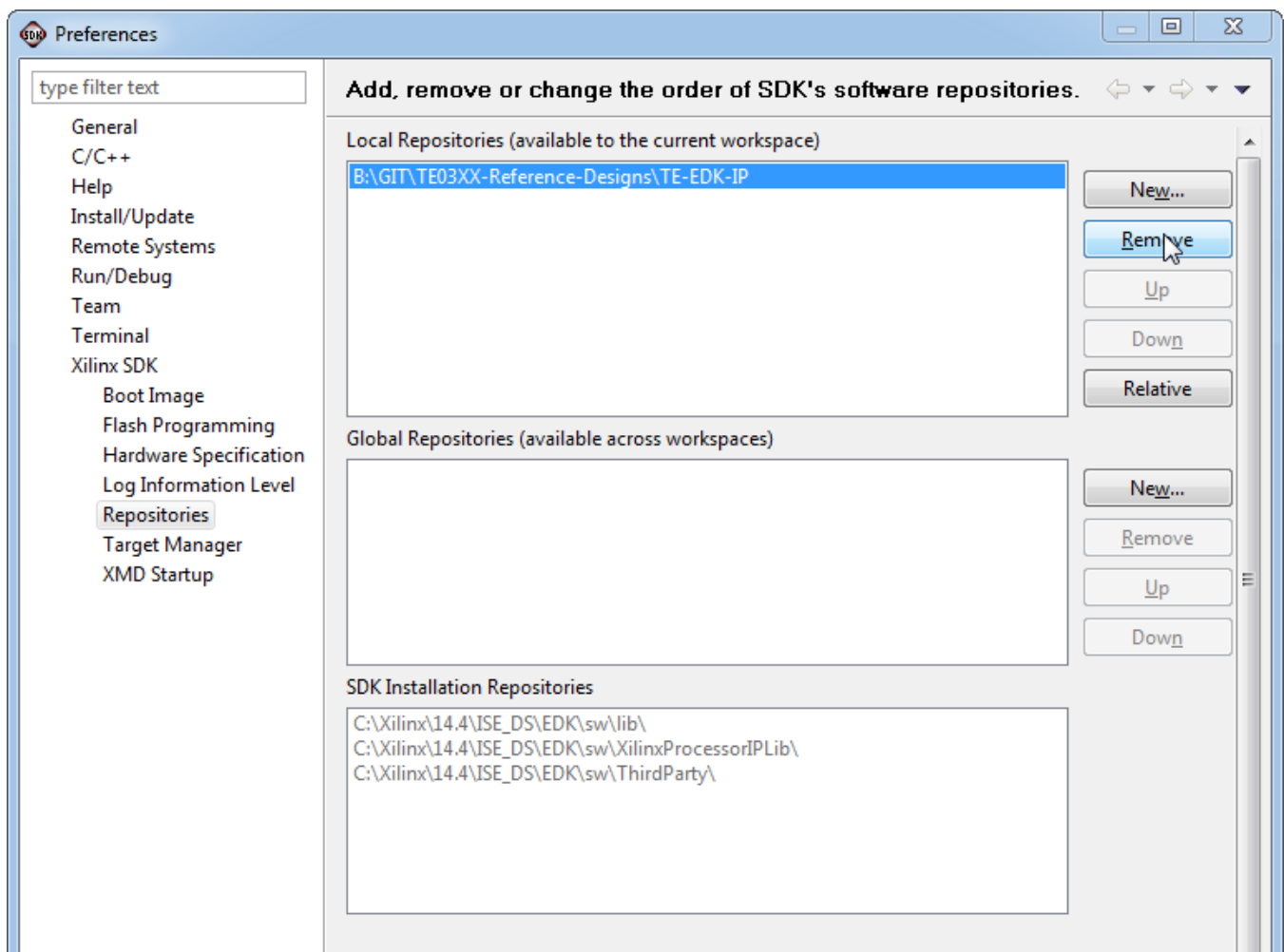
After this, even if refresh operation appears correct and no errors are reported, you should check the SDK Workspace references to TE EDK IP core drivers.

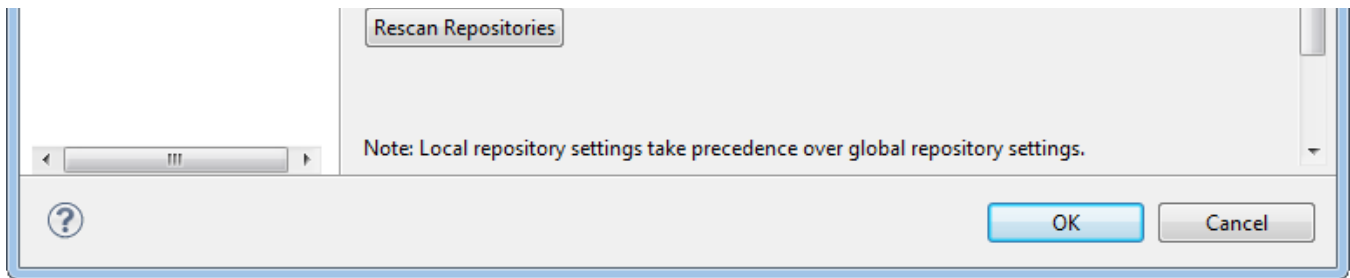
To open the SDK workspace reference repositories you should click "Xilinx Tools" > "Repositories".



Open Xilinx SDK Workspace repositories

A new pop-up windows will appear. You should remove the wrong reference clicking/selecting "B:\GIT\TE03XX-Reference-Designs\TE-EDK-IP" and clicking "Remove" button.

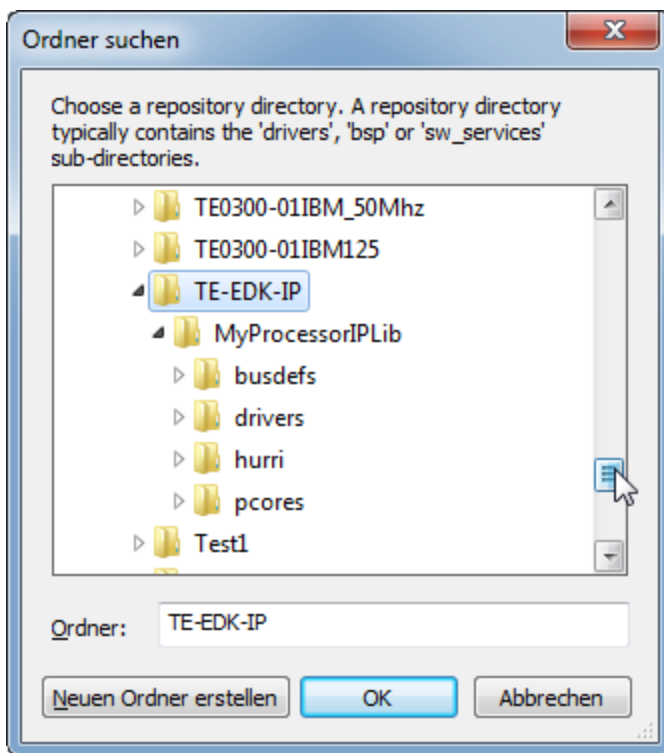




Preference pop-up, repositories selected

You should add the correct reference to TE EDK IP cores drivers.

Click the "New..." button and a pop-up "Select folder" will appear.

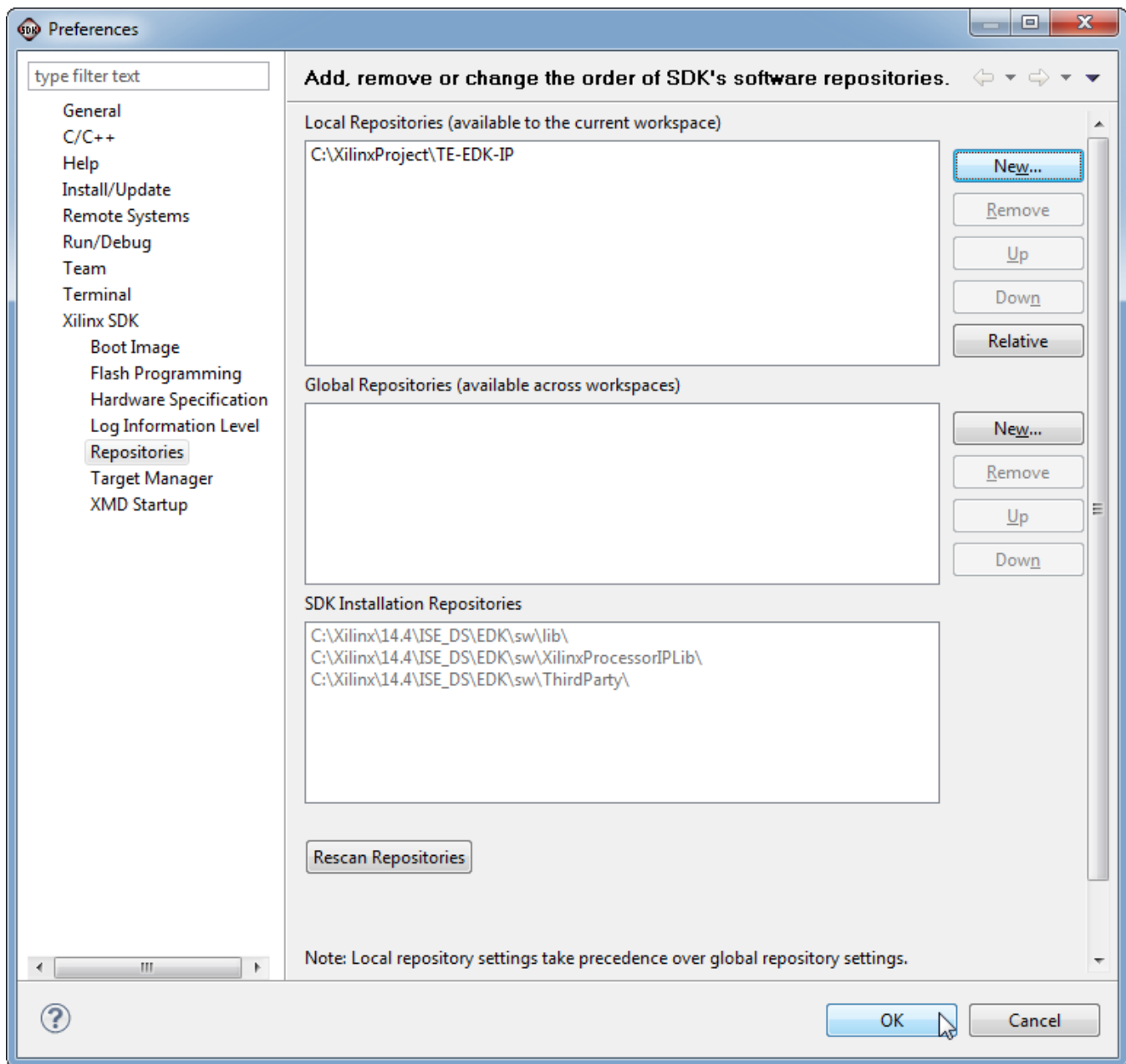


You should select the desired folder: "TE-EDK-IP" or "TE-EDK-IP\MyProcessorIPlib" should be both correct



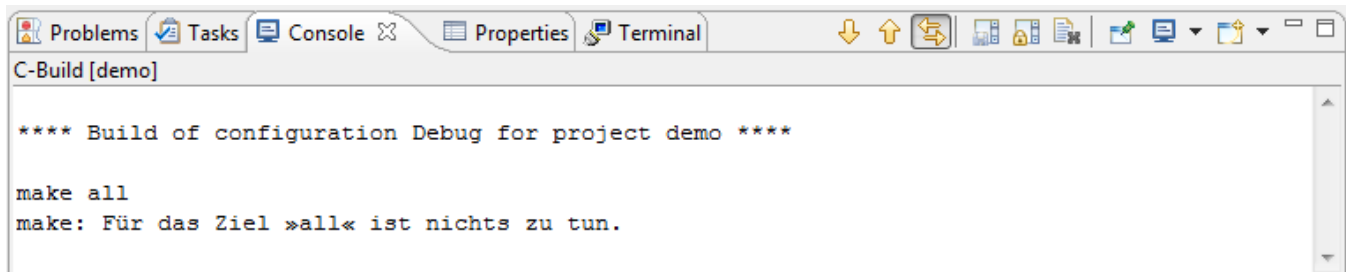
It is better if you do not alter folder nesting because double nesting of folders is a Xilinx XPS requirements (even if it is not an SDK requirement).

Then click "OK". A new SDK reference for IP Core drivers should appear.



TE EDK IP cores drivers references added

Then click "OK" button. The pop-up "Preferences" will close and you should wait until the following message appears.



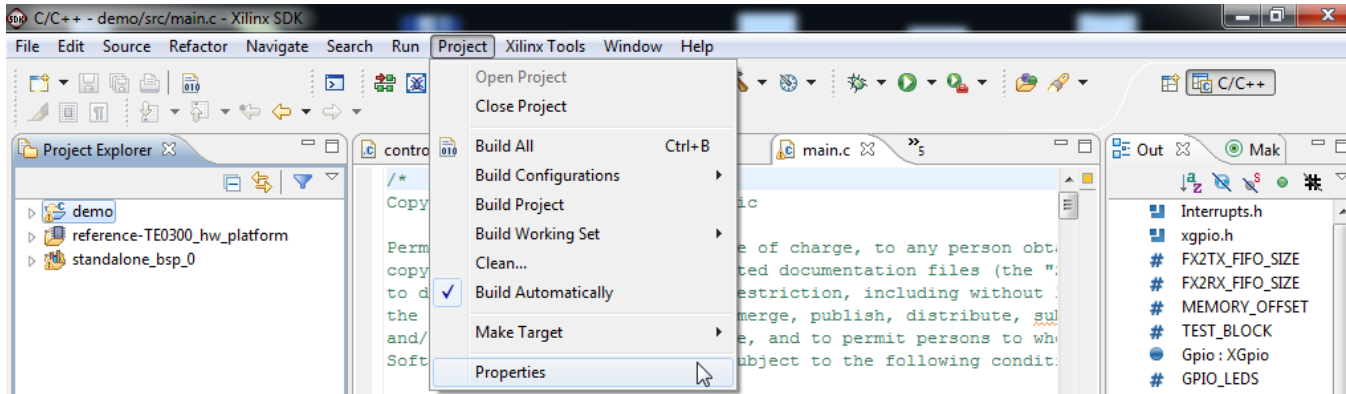
```
**** Build of configuration Debug for project demo ****

make all
make: Für das Ziel »all« ist nichts zu tun.
```

Build demo

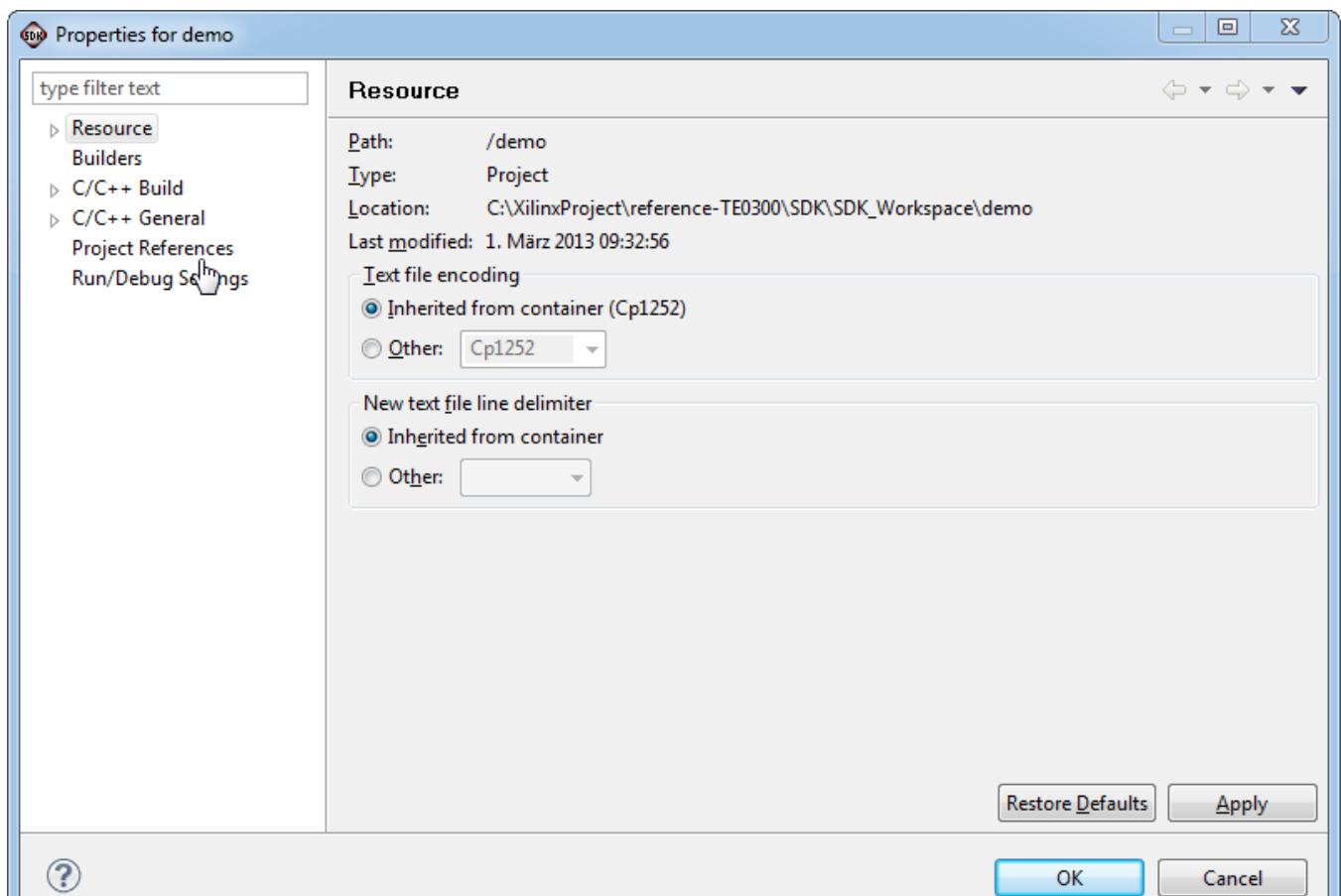
You should also check the "Application Project" "demo" references.

Click/select "demo" in "Project Explorer", then click "Project">"Properties".



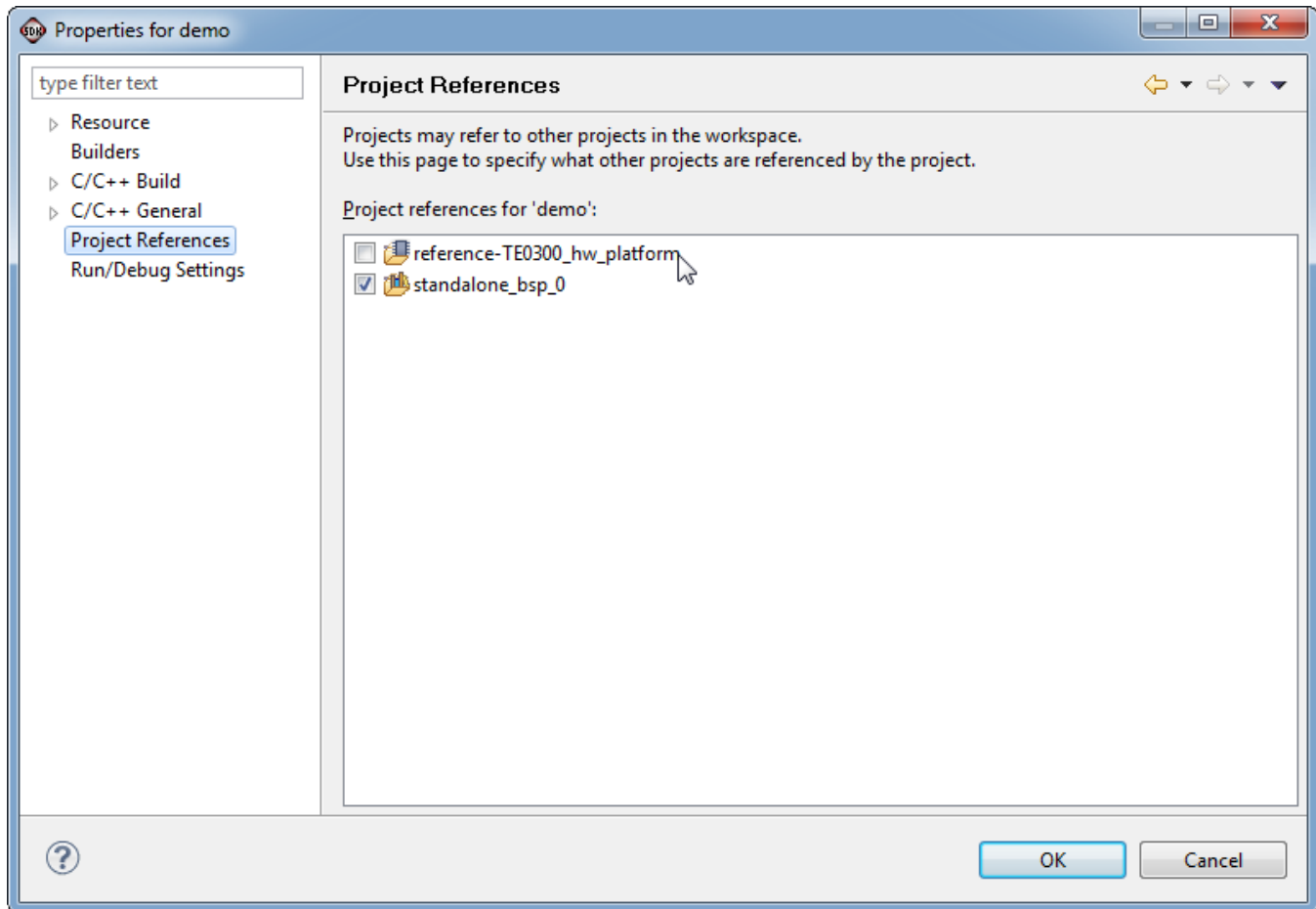
Open demo's project properties

The pop-up "Properties for demo" will open on "Resource"



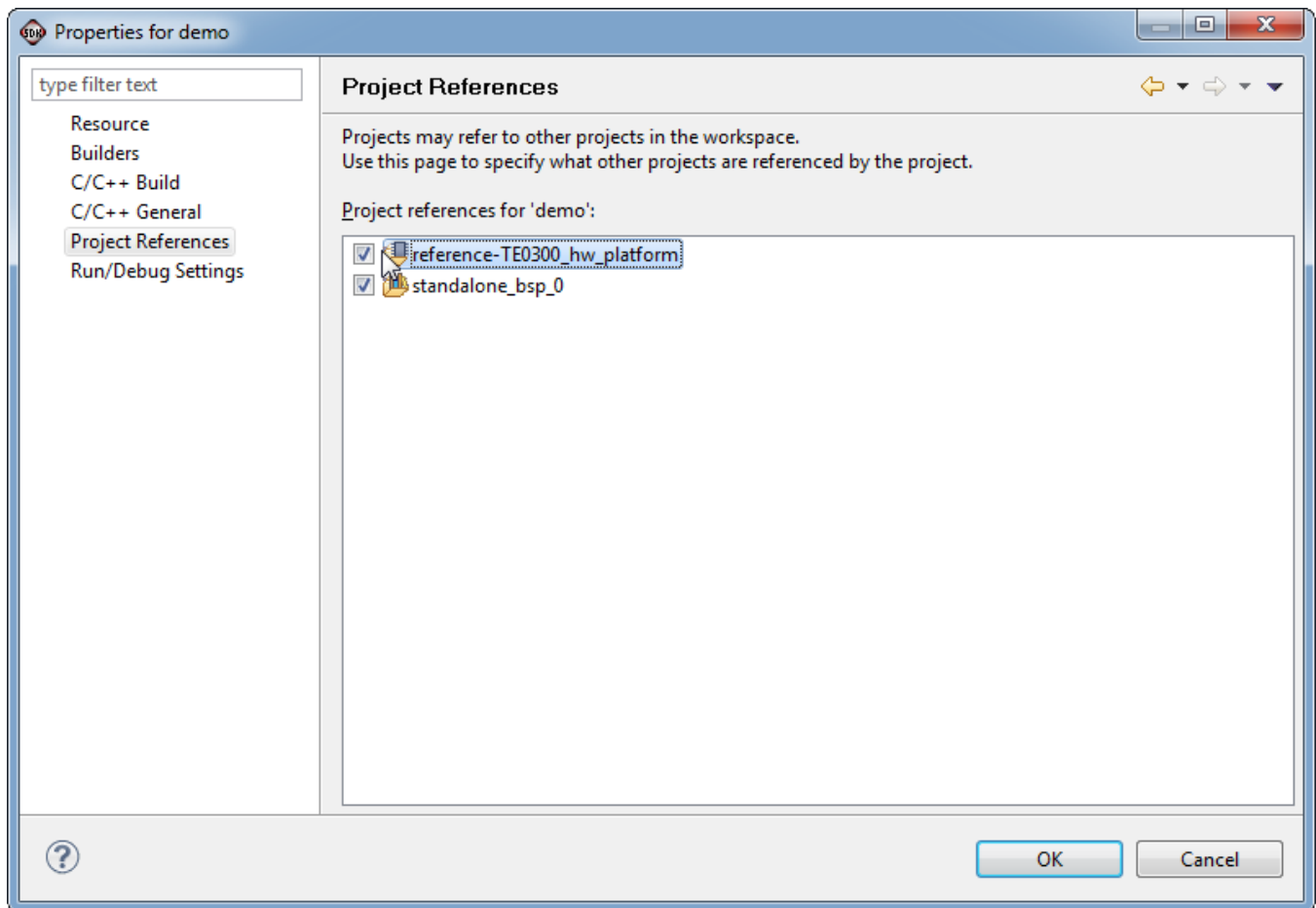
Demo's project properties opened

You should click /select "Project References". A new window will appear on the same pop-up.



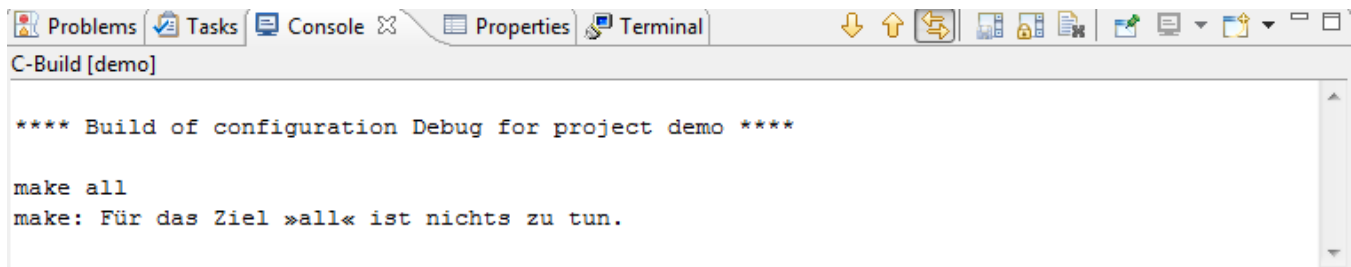
"Demo Project References" imported from original SDK workspace

You should also check (✓) "reference-TE0300_hw_platform".



"Demo Project References" modified

After this you should click "OK". You should wait until this message will appear.



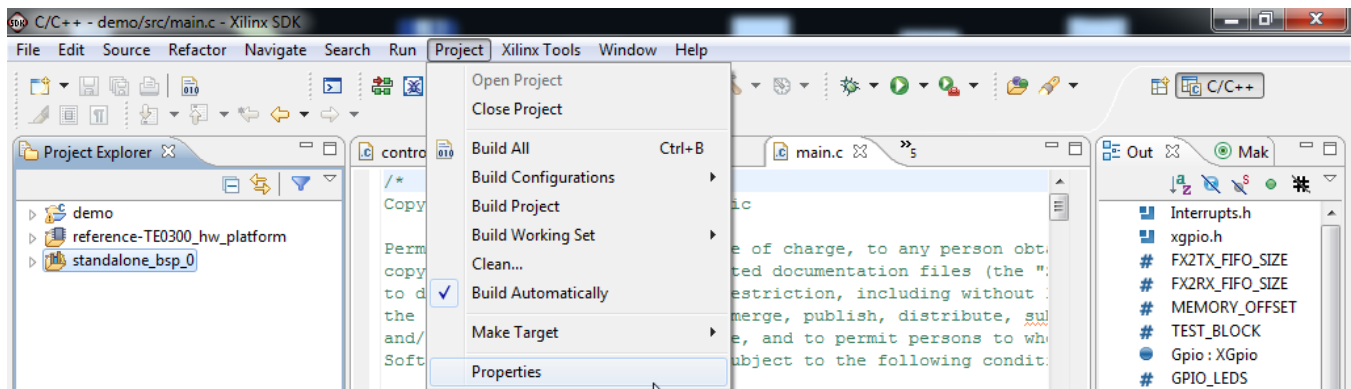
```
**** Build of configuration Debug for project demo ****

make all
make: Für das Ziel »all« ist nichts zu tun.
```

"Demo Project References" compile ends

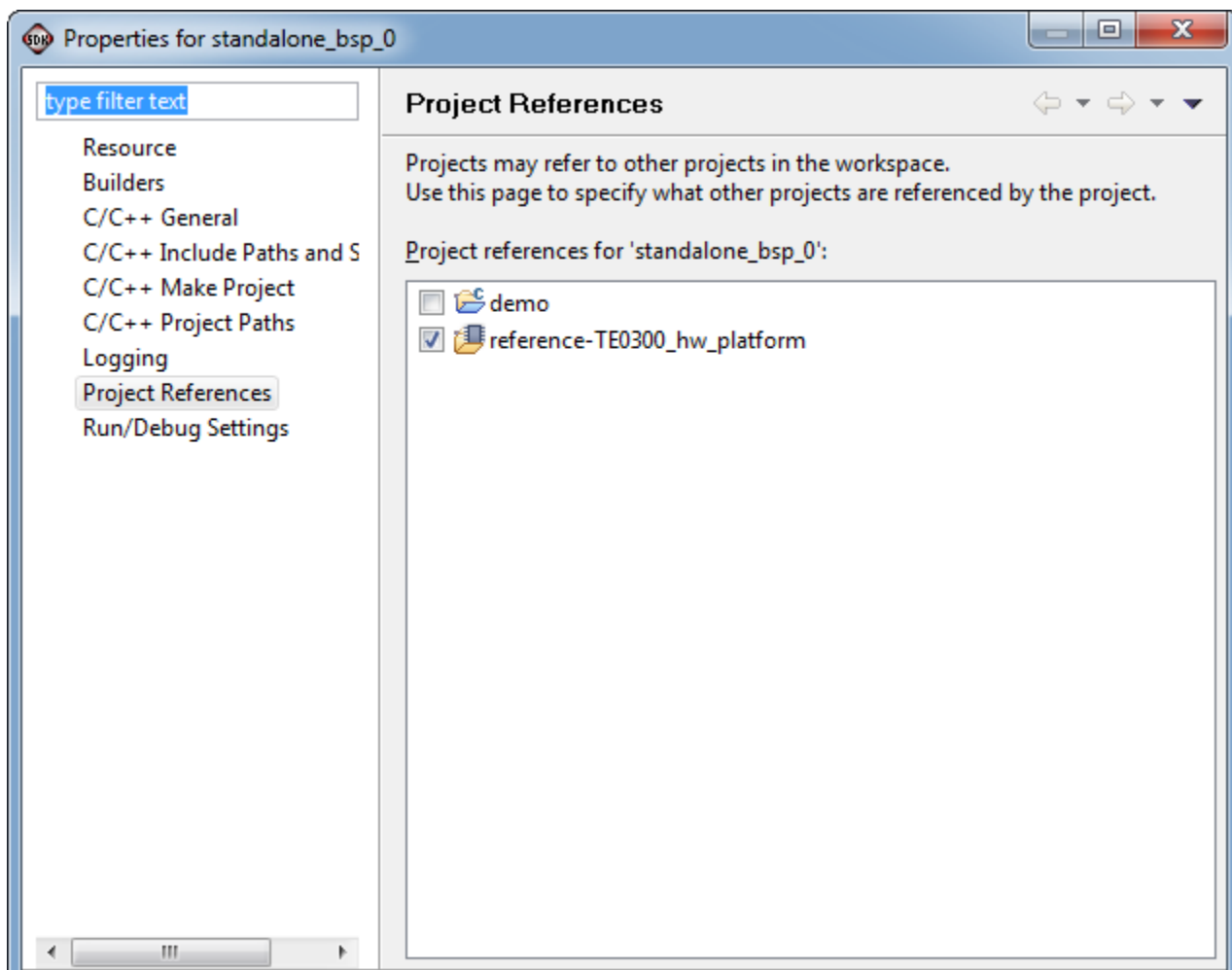
You should also check the "Board Support Package Project" references.

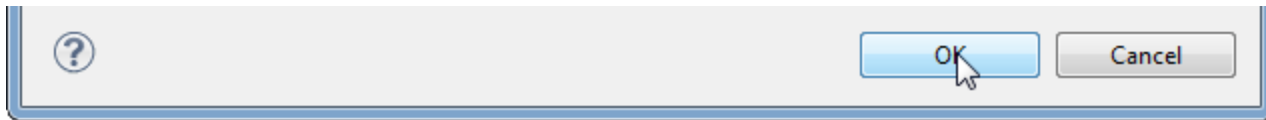
Click/select "standalone_bsp_0" in "Project Explorer", then click "Project">"Properties".



Open bsp's project properties

A new pop-up window "Properties for standalone_bsp_0" will appear.





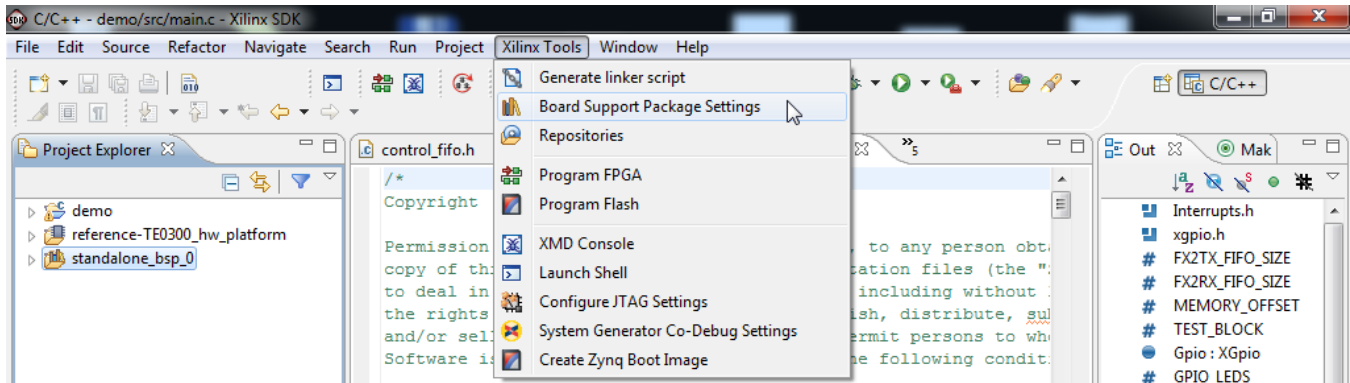
"standalone_bsp_0 project" references

In this case the bsp setting is correct. Only "reference-te0300_hw_platform" should be checked (✓).

Click "OK" and the pop-up will dissappear.

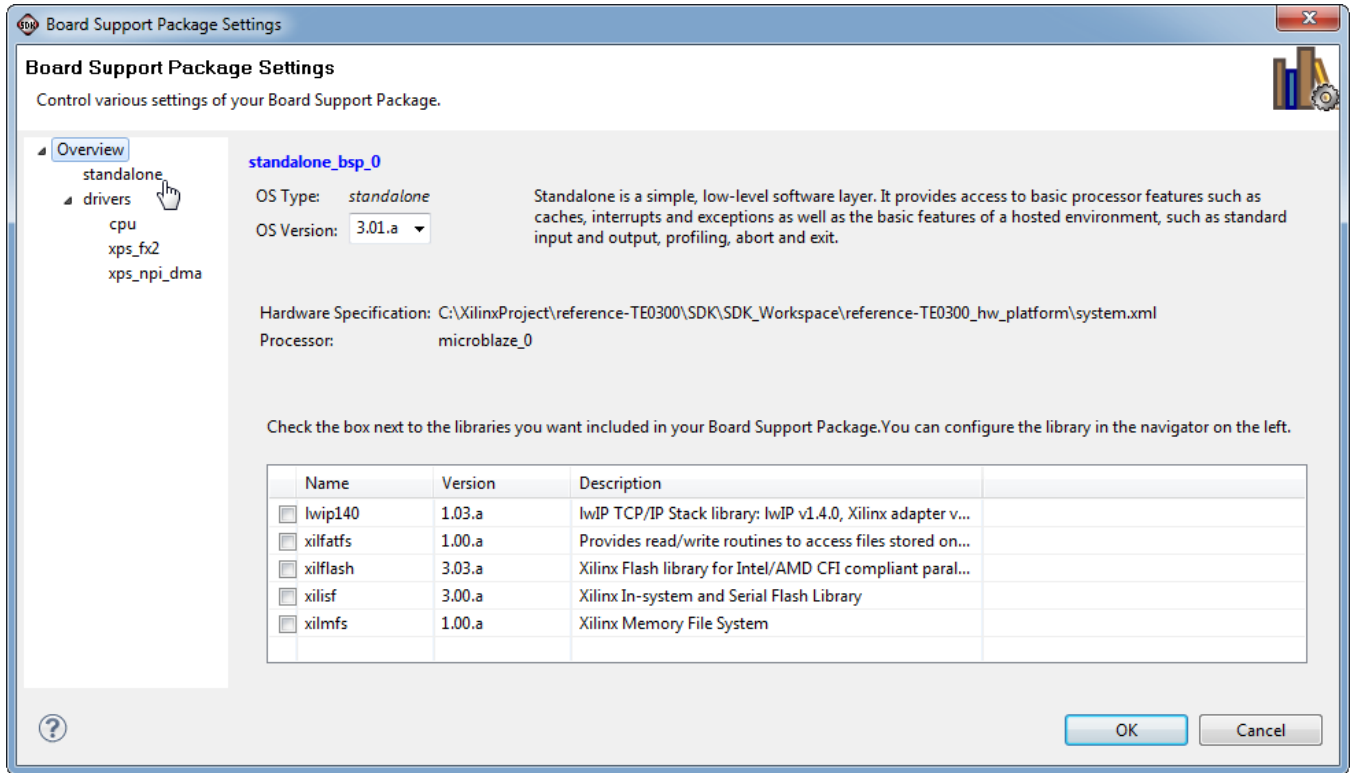
Now, you should select if you desire to debug on a real UART (rs232) or a simulated UART on JTAG (debug_module).

You should click/select "standalone_bsp_0" in "project Explorer", then you should click "Xilinx Tools" > "Board Support Package Settings".



Board Support Package Settings opening

A new pop-up will appear "Board Support Package Settings".



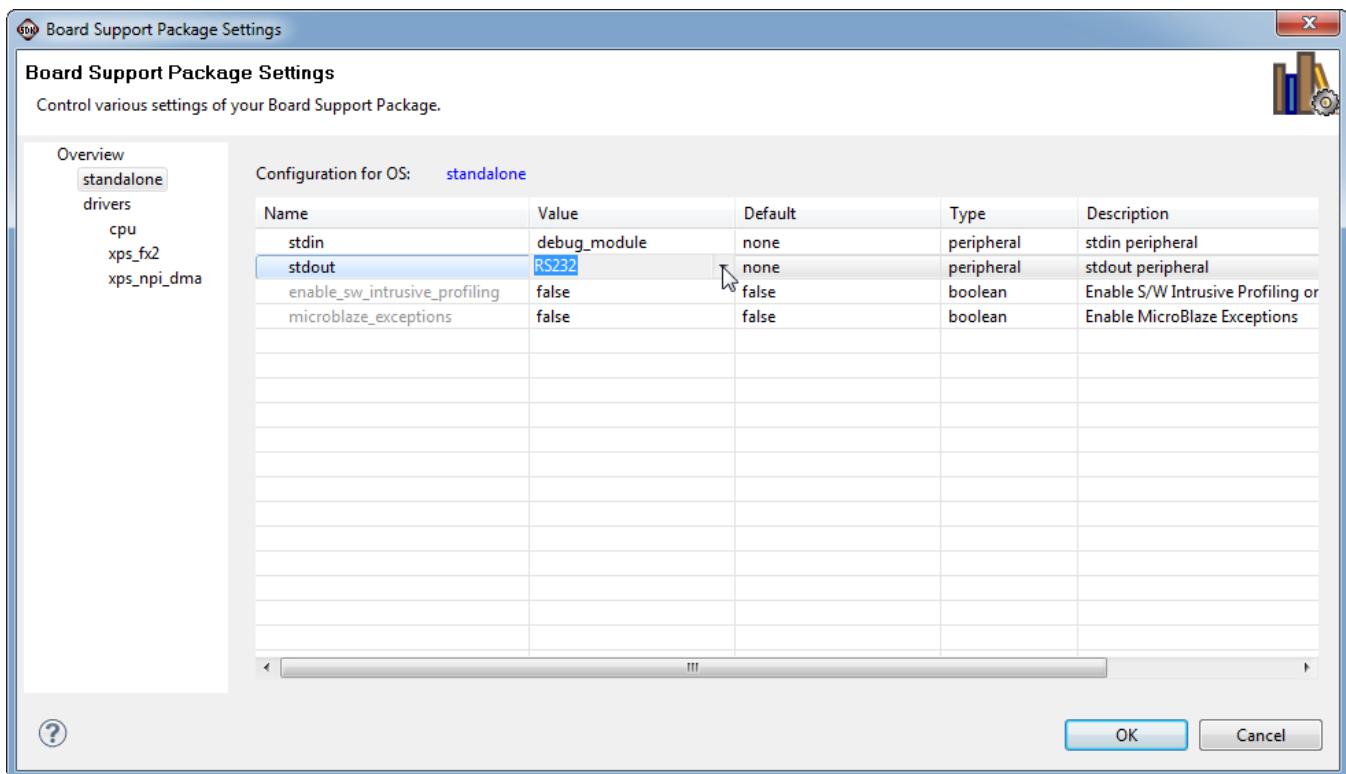
"Board Support Package Settings" pop-up

After this, you should click "standalone" and set "stdin" and "stdout" to "rs232" or "debug_module":

1. you should select "rs232" if you desire to use a real UART (and not a simulated UART though JTAG connection);
2. you should select "debug_module" if you desire that the XMD_UART works as local UART through the JTAG connection.

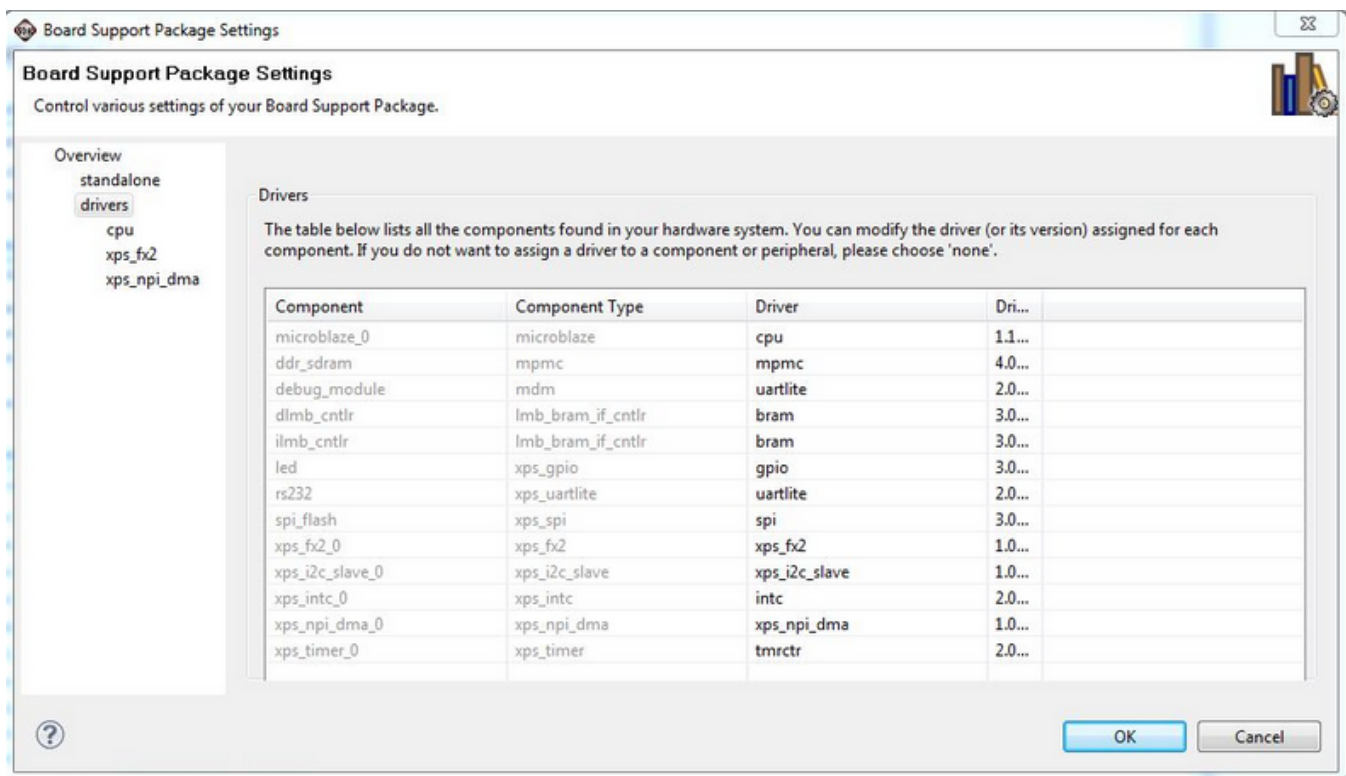


Demo program (running on MicroBlaze) will work even in case the UART port is left unconnected: it is not necessary to use a USB/Uart converter or Uart port on a PC, if you are using XMD UART HDL block



BSP settings: rs232 or debug_module

After this, you should click "drivers" to verify that all Microblaze components are supported by the driver in the repository "C:\XilinxProject\TE-EDK-IP".



BSP driver checking

Click "cpu". In particular you should also verify that "xmdstub_peripheral" is assigned to none if you desire to use debug_module instead of rs232.

