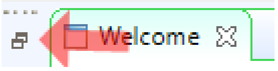
Vitis

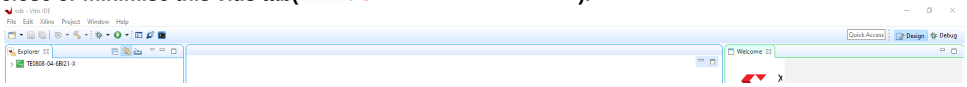
Table of content

Create Vitis Project from Vivado

New 2019.2 reference designs provide scripts to generate platform project with local repository for the given reference design.

After implementation/bitstream generation run on Vivado TCL console: **TE::sw_run_vitis**

- **Scripts generate Platform with the given article name of the project. The overview window will not closed by default in this case, please close or minimise this vitis tab(****).**

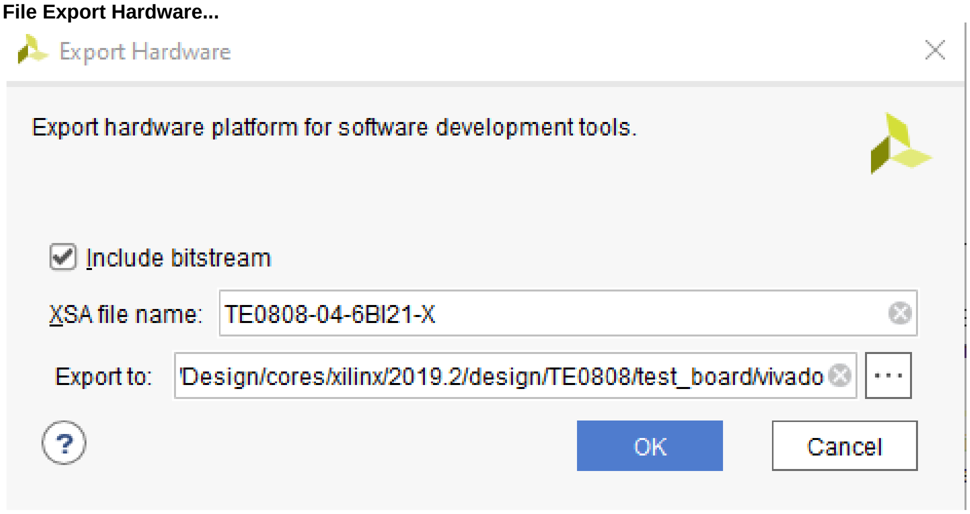


To create all manually see subsections below.

Create XSA with Vivado

This step must be done with Vivado. Synthesis (for xsa without bitstream) or implementation/bitstream step (for xsa with bitstream) must be done before.

- **File Export Hardware...**



Export Hardware

Export hardware platform for software development tools.

☑ Include bitstream

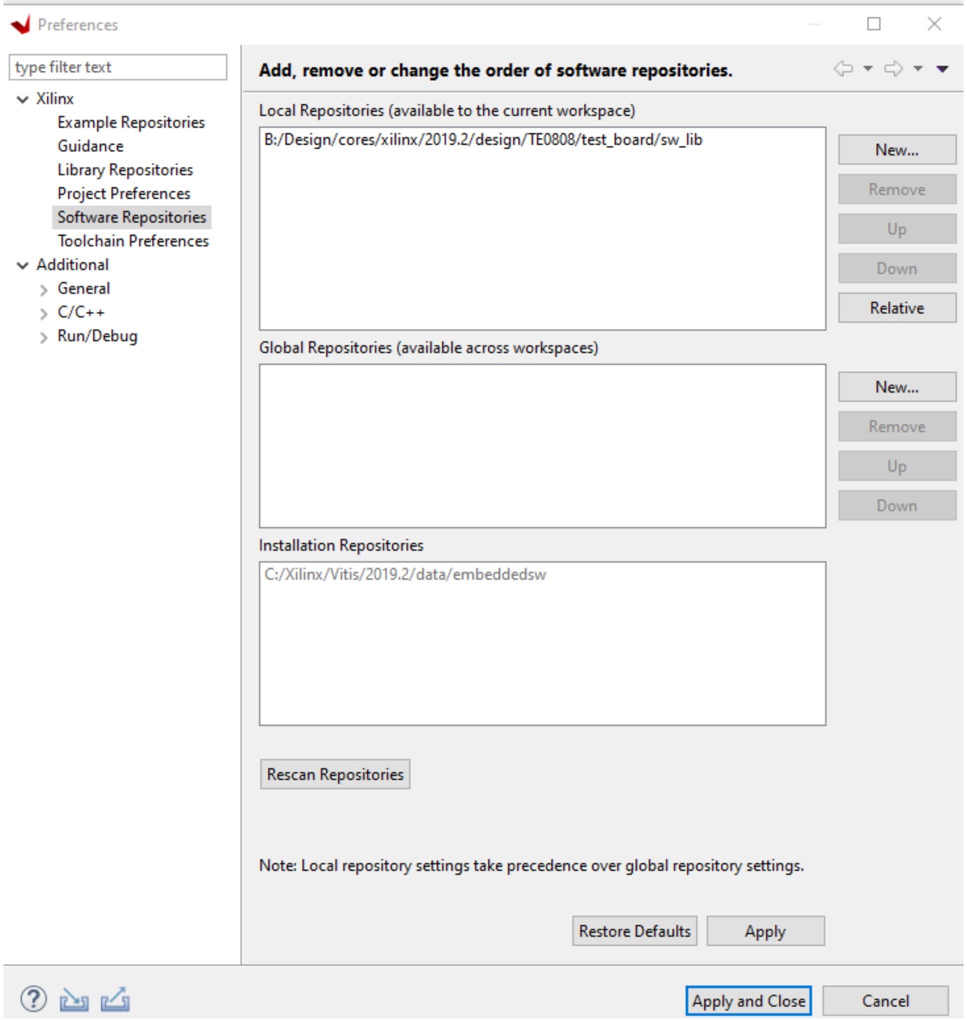
XSA file name: TE0808-04-6BI21-X

Export to: 'Design/cores/xilinx/2019.2/design/TE0808/test_board/vivado

Include local repositories

This step is need for local trenz SDK libraries. Add <reference design folder>/sw_lib to the repository_

- **Xilinx Repositories**



Add, remove or change the order of software repositories.

Local Repositories (available to the current workspace)

B:/Design/cores/xilinx/2019.2/design/TE0808/test_board/sw_lib

Global Repositories (available across workspaces)

Installation Repositories

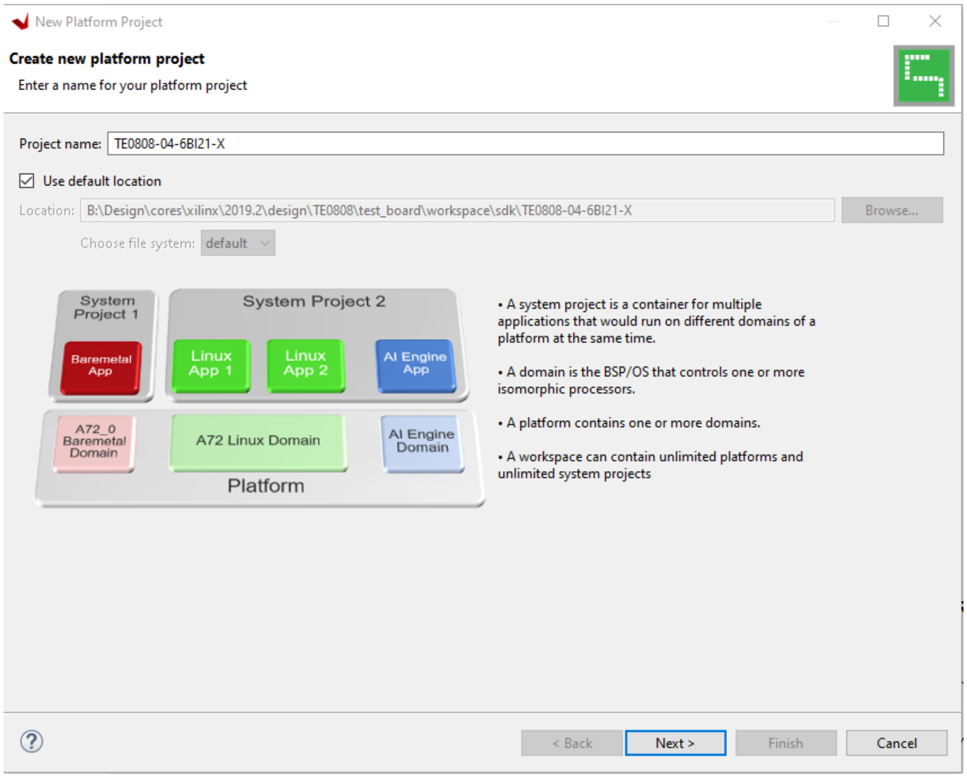
C:/Xilinx/Vitis/2019.2/data/embeddedsw

Note: Local repository settings take precedence over global repository settings.

Create Platform project for XSA

1. On Welcome Window press "**Create Platform Project**" or "**File New Platform Project**"

2. Add name



Create new platform project

Enter a name for your platform project

Project name: TE0808-04-6BI21-X

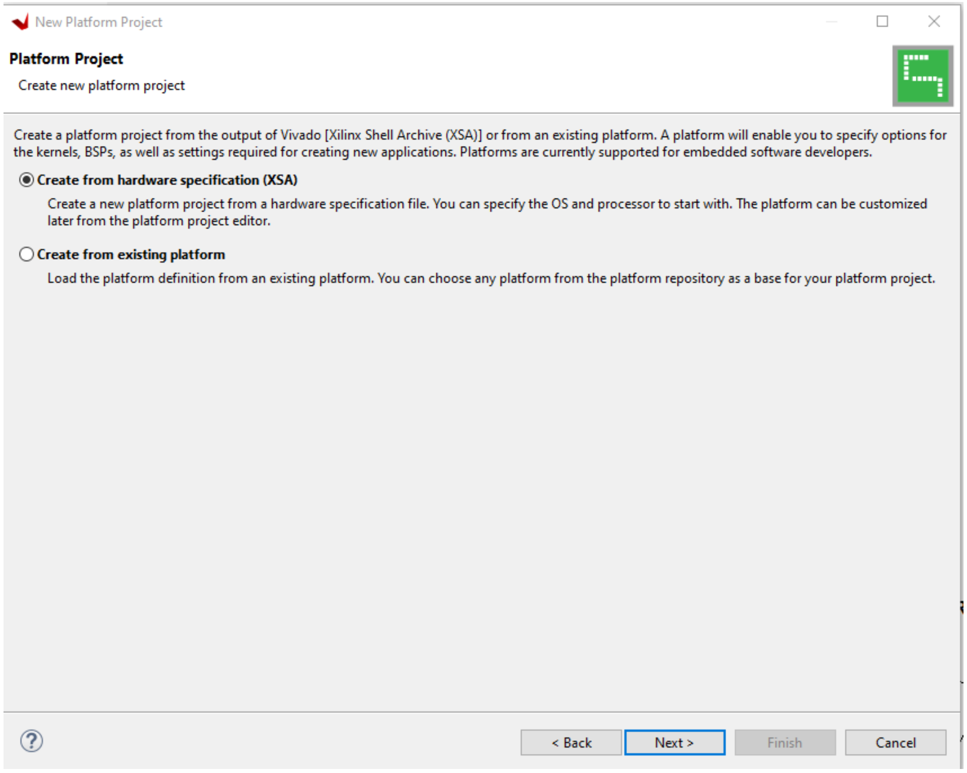
☑ Use default location

Location: B:\Design\cores\xilinx\2019.2\design\TE0808\test_board\workspace\sdk\TE0808-04-6BI21-X

Choose file system: default

- A system project is a container for multiple applications that would run on different domains of a platform at the same time.
- A domain is the BSP/OS that controls one or more isomorphic processors.
- A platform contains one or more domains.
- A workspace can contain unlimited platforms and unlimited system projects

3. Select Create from XSA



Platform Project

Create new platform project

Create a platform project from the output of Vivado [Xilinx Shell Archive (XSA)] or from an existing platform. A platform will enable you to specify options for the kernels, BSPs, as well as settings required for creating new applications. Platforms are currently supported for embedded software developers.

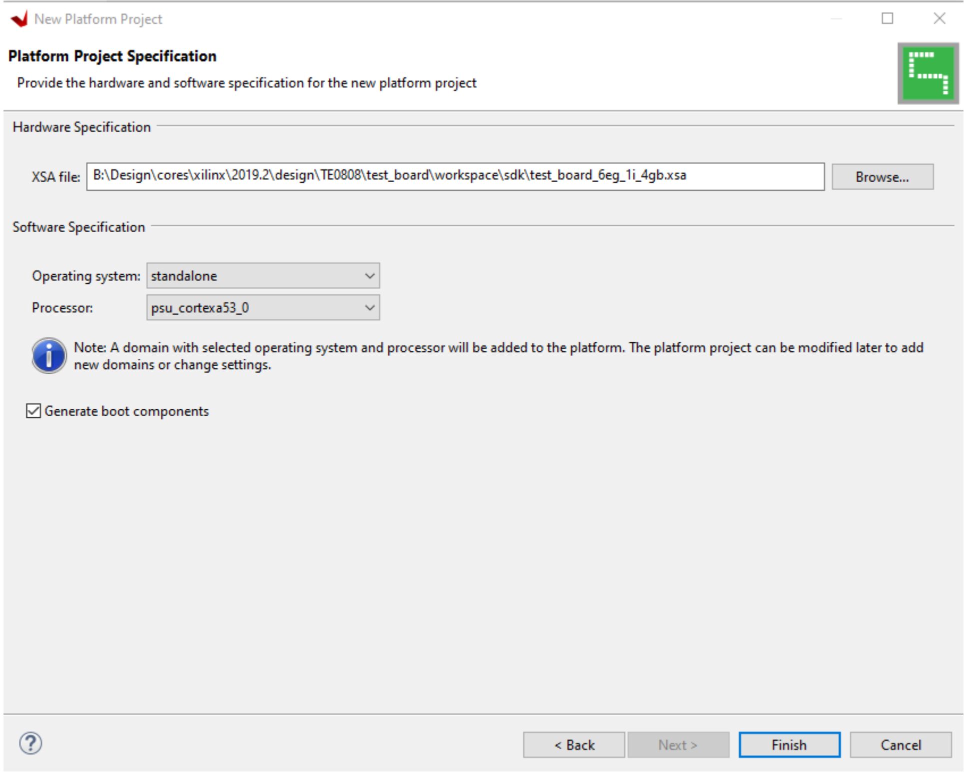
● **Create from hardware specification (XSA)**

Create a new platform project from a hardware specification file. You can specify the OS and processor to start with. The platform can be customized later from the platform project editor.

○ **Create from existing platform**

Load the platform definition from an existing platform. You can choose any platform from the platform repository as a base for your platform project.

4. Select XSA File
   Depending on usage: Change Operating system or Processor
   Recommended: Select Generate Boot components, which generates fsbl for Zynq or ZynqMP devices and pmufw for ZynqMP as separate domain into the Platform project



Platform Project Specification

Provide the hardware and software specification for the new platform project

Hardware Specification

XSA file: B:\Design\cores\xilinx\2019.2\design\TE0808\test_board\workspace\sdk\test_board_6eg_1i_4gb.xsa

Software Specification

Operating system: standalone

Processor: psu_cortexa53_0

Note: A domain with selected operating system and processor will be added to the platform. The platform project can be modified later to add new domains or change settings.

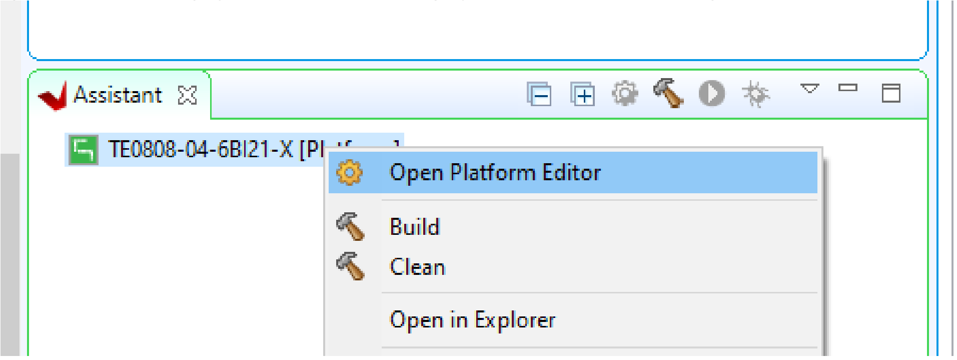
☑ Generate boot components

Create Domain

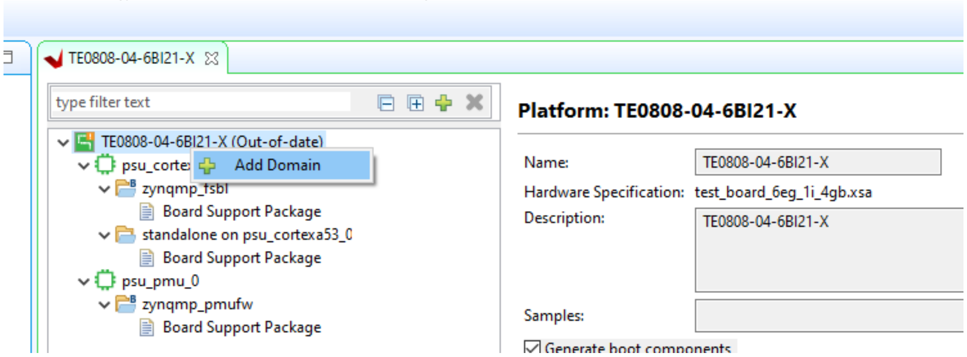
- Following Domains will be generated with Platform Project generation (if selected):
  - Stand-alone Domain for Stand-alone application (default selected on platform generation step)
  - FSBL Domain: for Zynq and ZynqMP devices
  - PMU Domain: for ZynqMP devices

Add new domain:

1. Open Platform Editor (Right click on the Platform project in the Assistant Window)



2. Add Domain (go the the Platform Editor Window)



Platform: TE0808-04-6BI21-X

Name: TE0808-04-6BI21-X

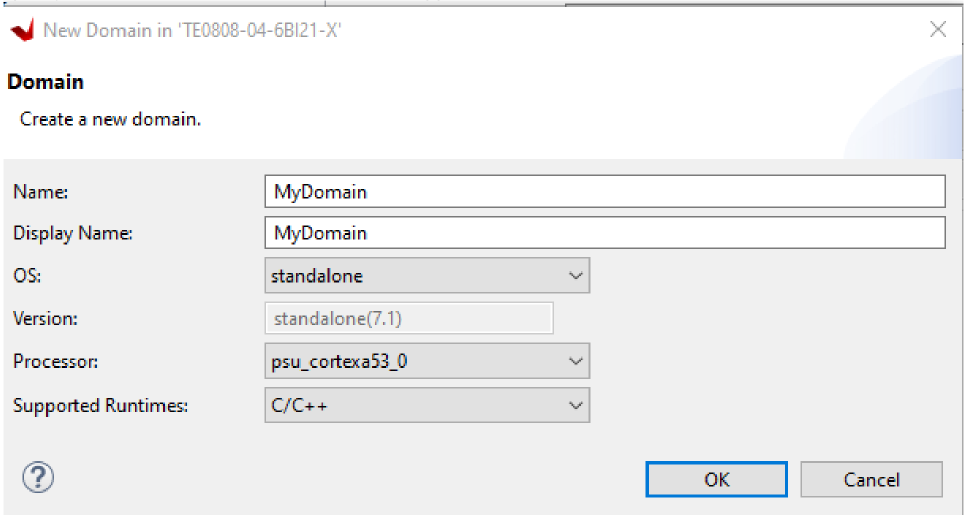
Hardware Specification: test_board_6eg_1i_4gb.xsa

Description: TE0808-04-6BI21-X

Samples:

☑ Generate boot components

3. Specify Name and Select OS, Processor and supported runtime:



New Domain in 'TE0808-04-6BI21-X'

Domain

Create a new domain.

Name: MyDomain

Display Name: MyDomain

OS: standalone

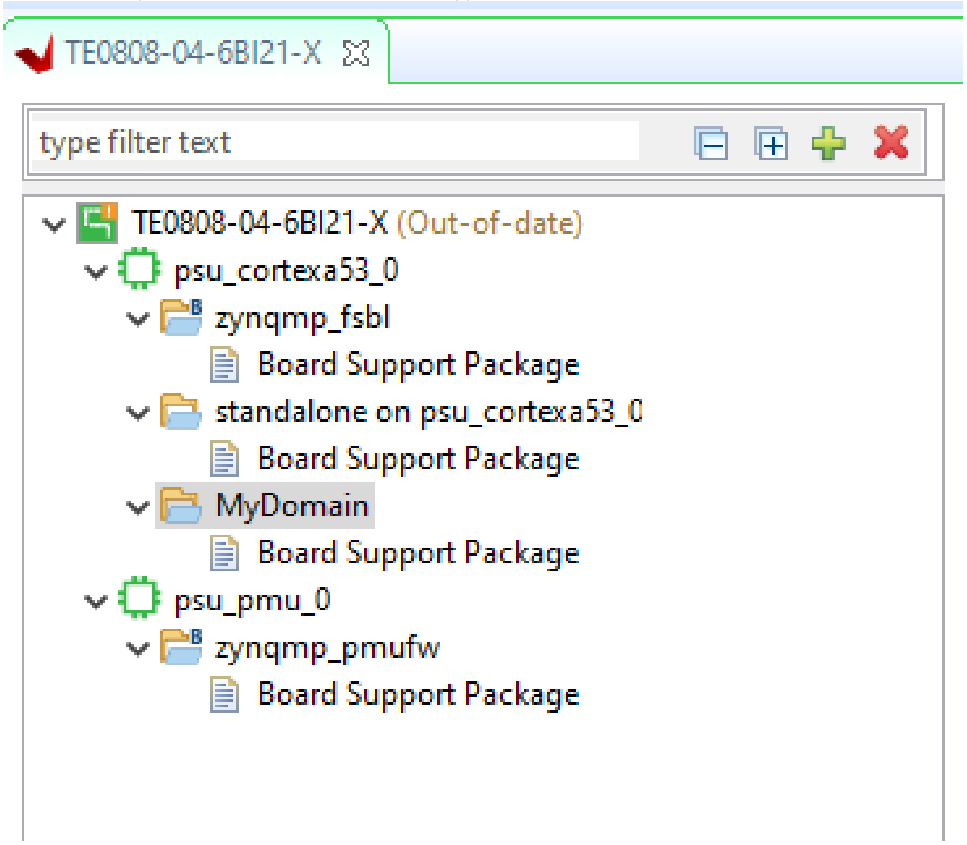
Version: standalone(7.1)

Processor: psu_cortexa53_0

Supported Runtimes: C/C++

Note: for Linux additional files must be add (BIF, boot files(u-boot.elf for Zynq and ZynqMP, atf firmware (bl31.elf) for ZynqMP) and linux image (image.ub) from petalinux,)

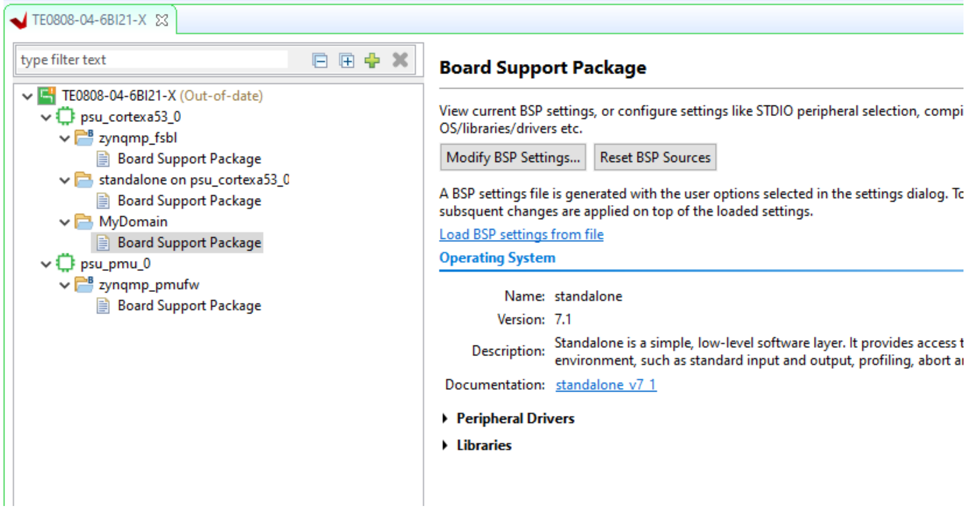
4. Domain will be generated and can be selected for applications:



Modify BSP Settings of the different Domains

Domain BSP modification are need to add for example Xilinx libs or modify BSP setup

1. On Platform Editor got the the Board Support Package of the Domain which should be modified and press "Modify BSP Settings..."



View current BSP settings, or configure settings like STDIO peripheral selection, compi
OS/libraries/drivers etc.

A BSP settings file is generated with the user options selected in the settings dialog. Tc
subsquent changes are applied on top of the loaded settings.

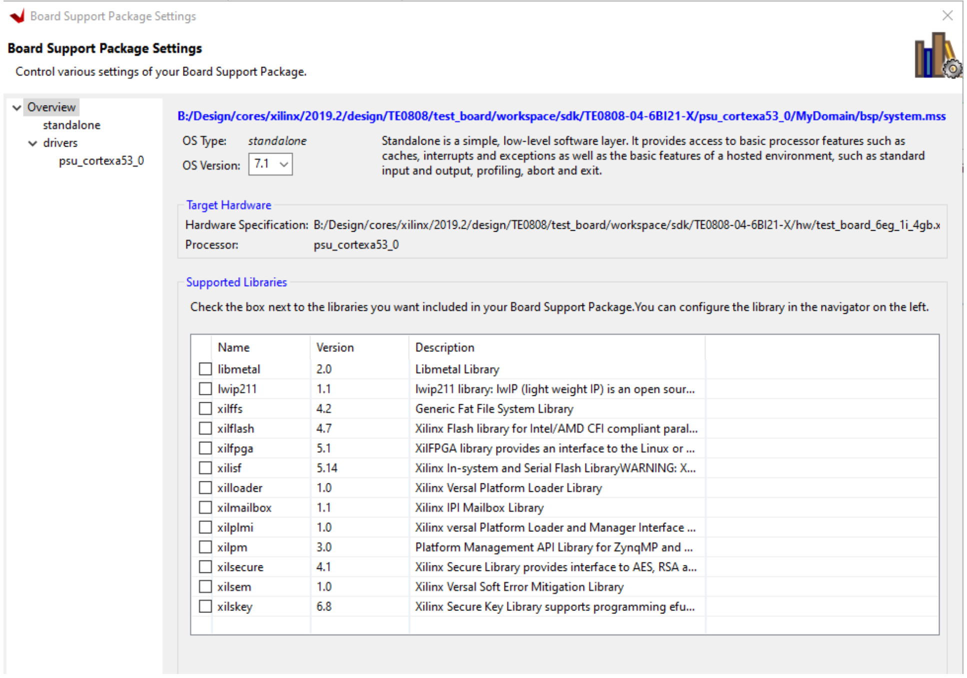
Name: standalone

Version: 7.1

Description: Standalone is a simple, low-level software layer. It provides access t
environment, such as standard input and output, profiling, abort ar

Documentation: standalone_v7_1

2. Add need drivers or change default BSP setup



Board Support Package Settings

Control various settings of your Board Support Package.

B:/Design/cores/xilinx/2019.2/design/TE0808/test_board/workspace/sdk/TE0808-04-6BI21-X/psu_cortexa53_0/MyDomain/bsp/system.mss

OS Type: *standalone*

OS Version: 7.1

Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.

Target Hardware

Hardware Specification: B:/Design/cores/xilinx/2019.2/design/TE0808/test_board/workspace/sdk/TE0808-04-6BI21-X/hw/test_board_6eg_1i_4gb.x

Processor: psu_cortexa53_0

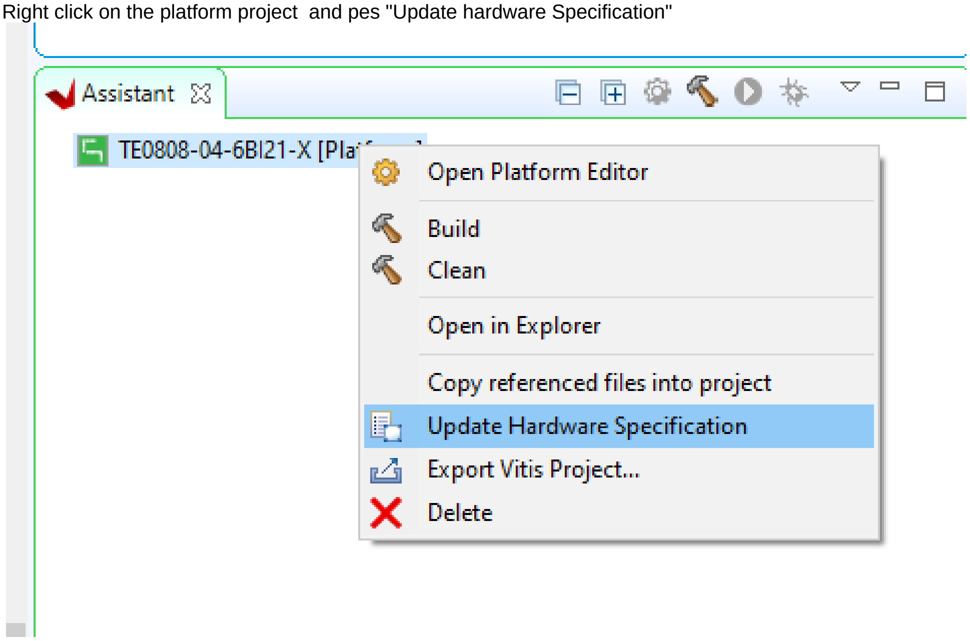
Supported Libraries

Check the box next to the libraries you want included in your Board Support Package.You can configure the library in the navigator on the left.

	Name	Version	Description
☐	libmetal	2.0	Libmetal Library
☐	lwip211	1.1	lwip211 library: lwIP (light weight IP) is an open sour...
☐	xilffs	4.2	Generic Fat File System Library
☐	xilflash	4.7	Xilinx Flash library for Intel/AMD CFI compliant paral...
☐	xilfpga	5.1	XilFPGA library provides an interface to the Linux or ...
☐	xilisf	5.14	Xilinx In-system and Serial Flash LibraryWARNING: X...
☐	xilloader	1.0	Xilinx Versal Platform Loader Library
☐	xilmailbox	1.1	Xilinx IPI Mailbox Library
☐	xilplmi	1.0	Xilinx versal Platform Loader and Manager Interface ...
☐	xilpm	3.0	Platform Management API Library for ZynqMP and ...
☐	xilsecure	4.1	Xilinx Secure Library provides interface to AES, RSA a...
☐	xilsem	1.0	Xilinx Versal Soft Error Mitigation Library
☐	xilskey	6.8	Xilinx Secure Key Library supports programming efu...

Update Platform with modified XSA

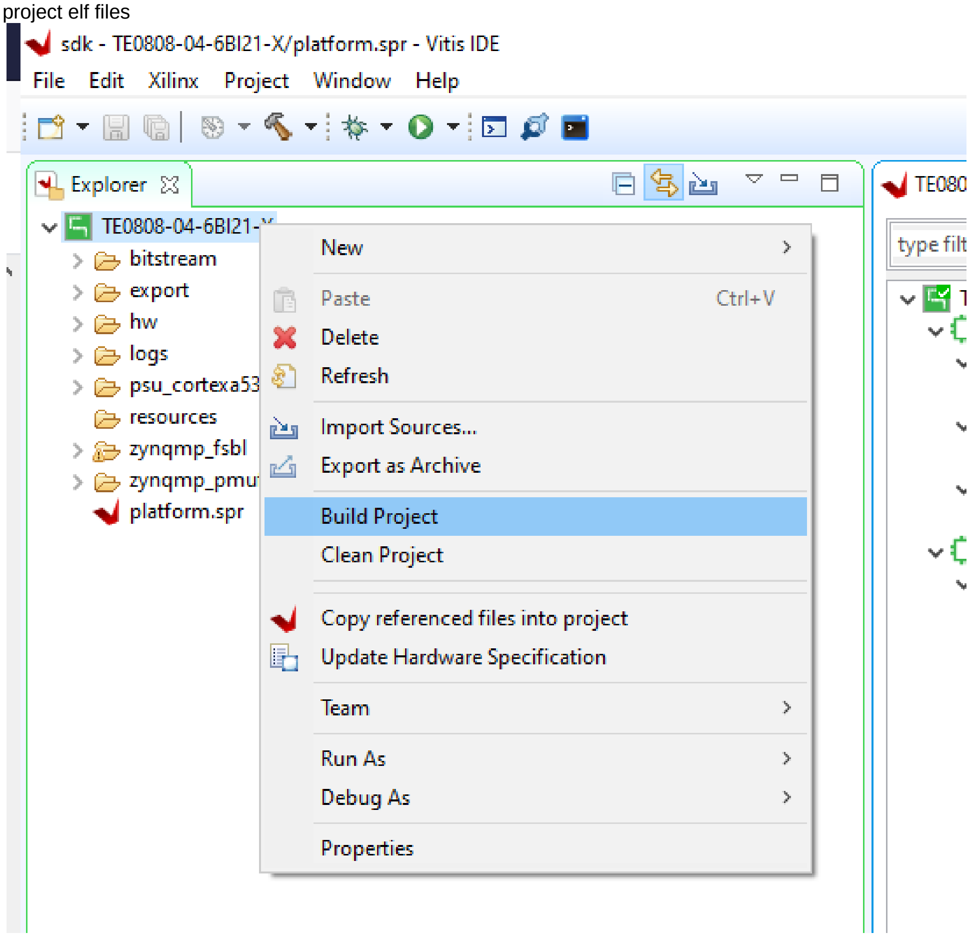
1. Right click on the platform project and pes "Update hardware Specification"



Note: if update works depends on the changes in the XSA file (your Vivado PS setup), it's saver to create a new vitis platform project with the new XSA!

Build Platform Project

1. Right click on the platform project in the Explore Window and press "**Build Project**" or "**Clean Project**" to generate/regenerate the platform project elf files



Bare-metal Software Application

Applications which are provided with the reference designs, can be generated directly with vivado. After implementation/bitstream generation run on Vivado TCL console: **TE::sw_run_vitis -all**

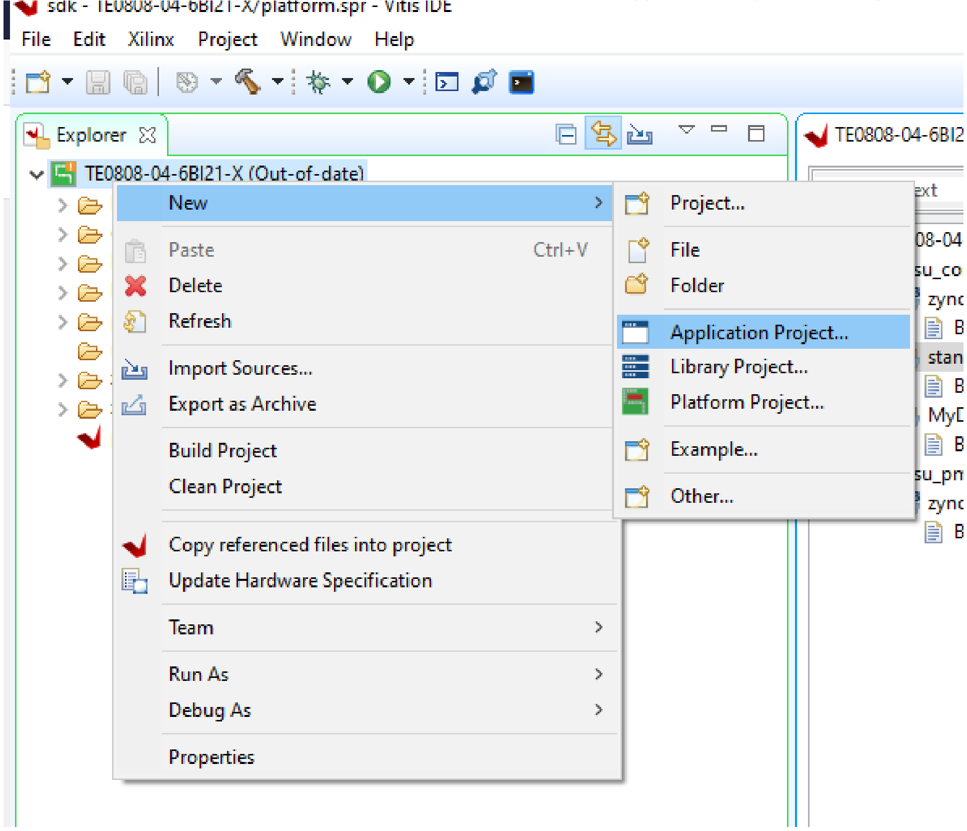
- **Scripts generate Platform and application which are defined in <reference design base folder>\sw_lib\apps-list.csv. App generation needs some more time to finished(depends on PC performance). The overview window will not closed by default in this case, please close or minimise this vitis tab(** **).**

Bare-metal application need a stand-alone domain, which must be generated at first (see CreateDomain). Stand-alone domain will be generated default with platform generation.

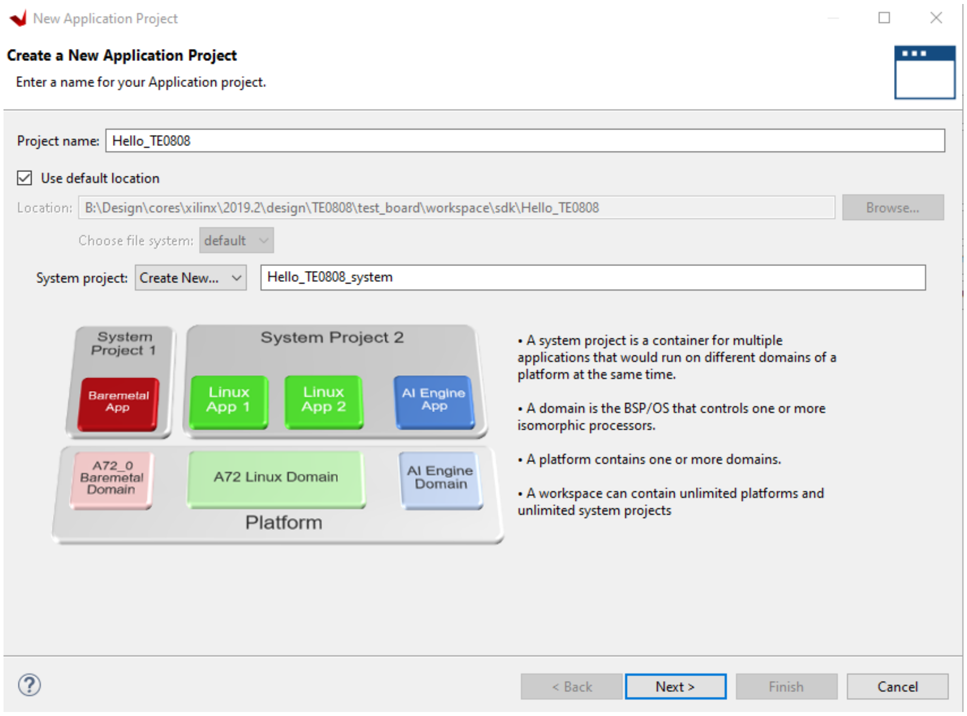
For manual bare metal project generation, see subsections below.

Create Application

1. Right click on the platform project in the Explore window or "File New Application Project" on the top menue



2. Set Name:



Create a New Application Project

Enter a name for your Application project.

Project name: Hello_TE0808

☑ Use default location

Location: B:\Design\cores\xilinx\2019.2\design\TE0808\test_board\workspace\sdk\Hello_TE0808

Choose file system: default

System project: Create New... Hello_TE0808_system

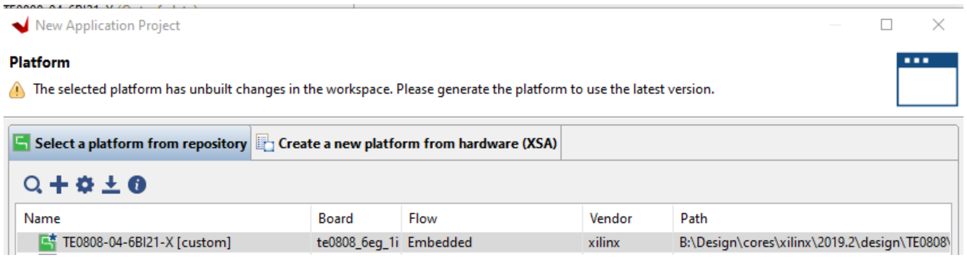
• A system project is a container for multiple applications that would run on different domains of a platform at the same time.

• A domain is the BSP/OS that controls one or more isomorphic processors.

• A platform contains one or more domains.

• A workspace can contain unlimited platforms and unlimited system projects

3. Select Platform

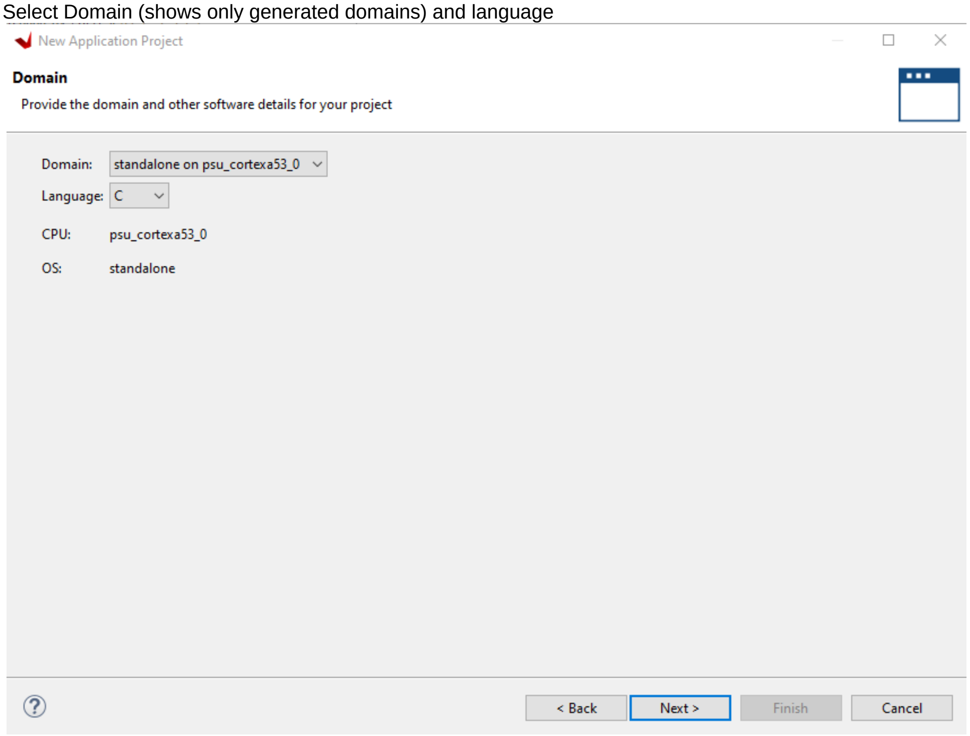

New Application Project
Select a platform from repository
Create a new platform from hardware (XSA)

Platform

⚠ The selected platform has unbuilt changes in the workspace. Please generate the platform to use the latest version.

Name	Board	Flow	Vendor	Path
TE0808-04-6BI21-X [custom]	te0808_6eg_1i	Embedded	xilinx	B:\Design\cores\xilinx\2019.2\design\TE0808\

4. Select Domain (shows only generated domains) and language



Domain

Provide the domain and other software details for your project

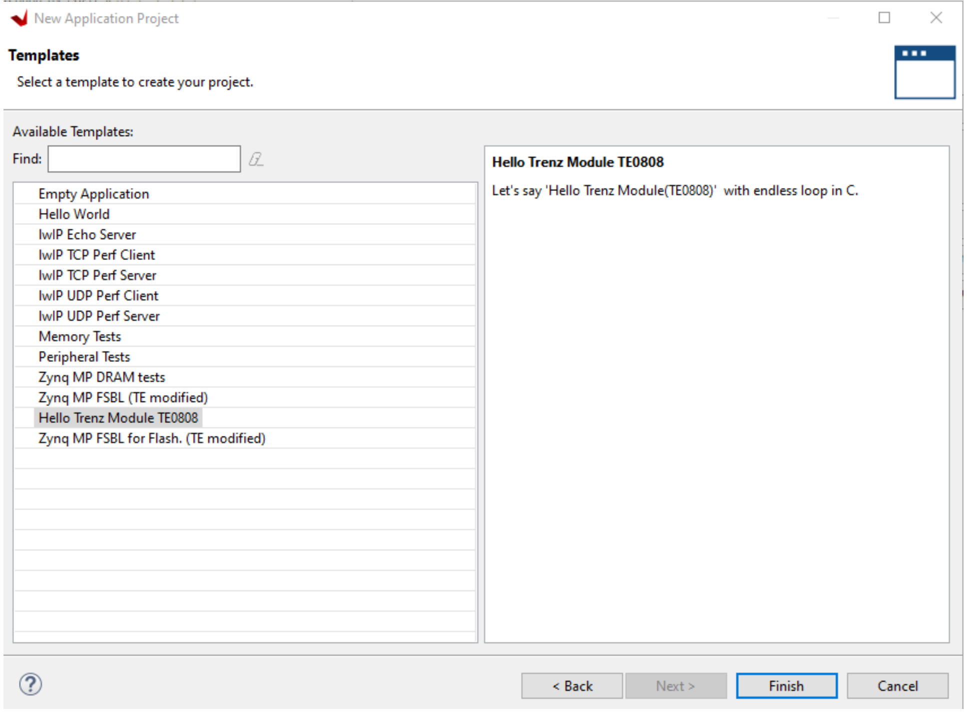
Domain: standalone on psu_cortexa53_0

Language: C

CPU: psu_cortexa53_0

OS: standalone

5. Select application



New Application Project

Templates

Select a template to create your project.

Available Templates:

Find:

- Empty Application
- Hello World
- lwIP Echo Server
- lwIP TCP Perf Client
- lwIP TCP Perf Server
- lwIP UDP Perf Client
- lwIP UDP Perf Server
- Memory Tests
- Peripheral Tests
- Zynq MP DRAM tests
- Zynq MP FSBL (TE modified)
- Hello Trenz Module TE0808
- Zynq MP FSBL for Flash. (TE modified)

Hello Trenz Module TE0808

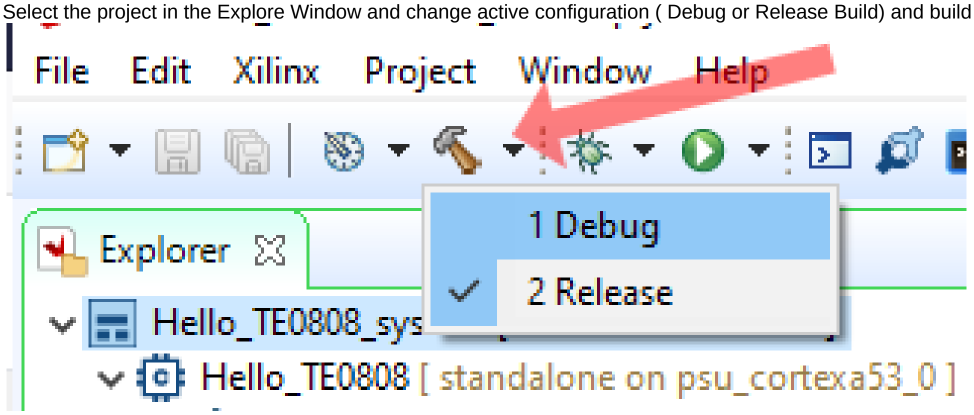
Let's say 'Hello Trenz Module(TE0808)' with endless loop in C.

Note: In case application needs some Xilinx libraries (shown as note, in case the app can't be selected for generation), add these libraries to the domain BSP before you try to generate the application.

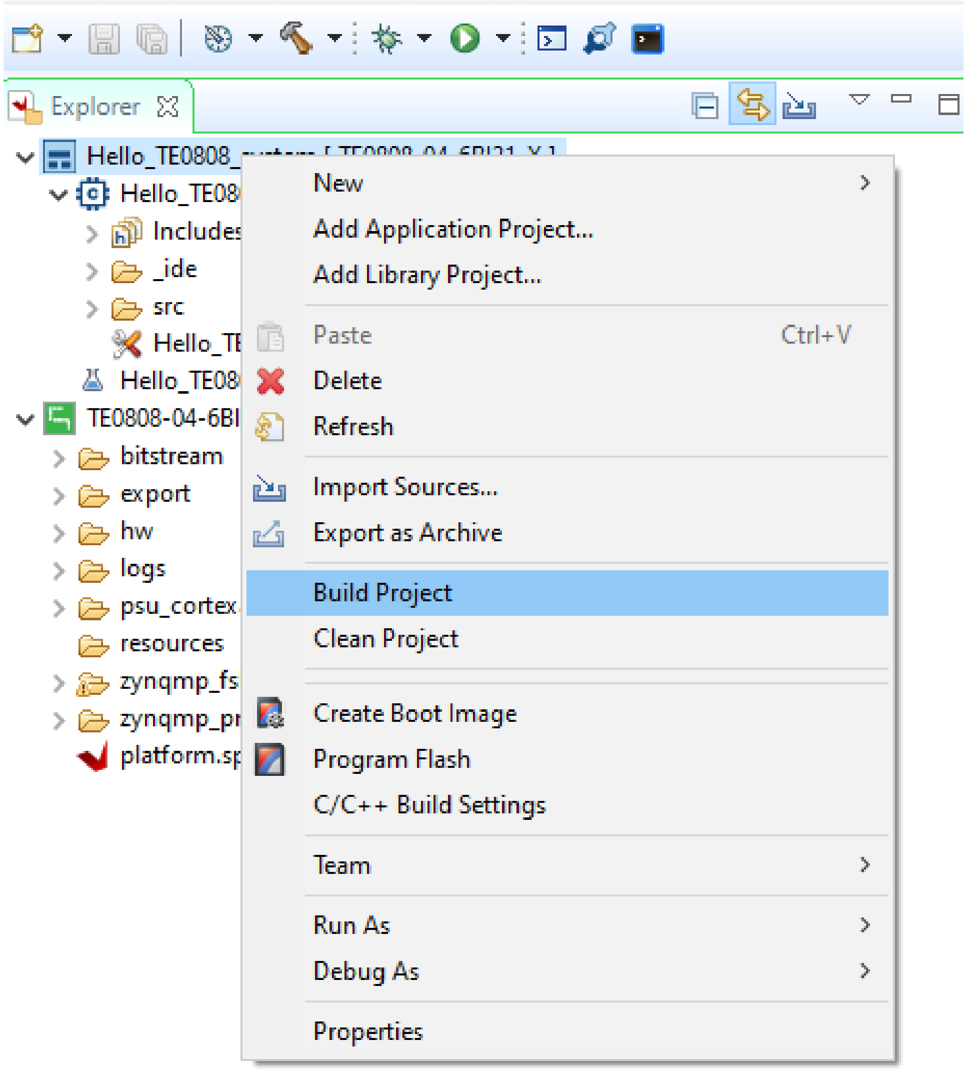
Build Application

Important: Build Platform Project, in case it's not done before

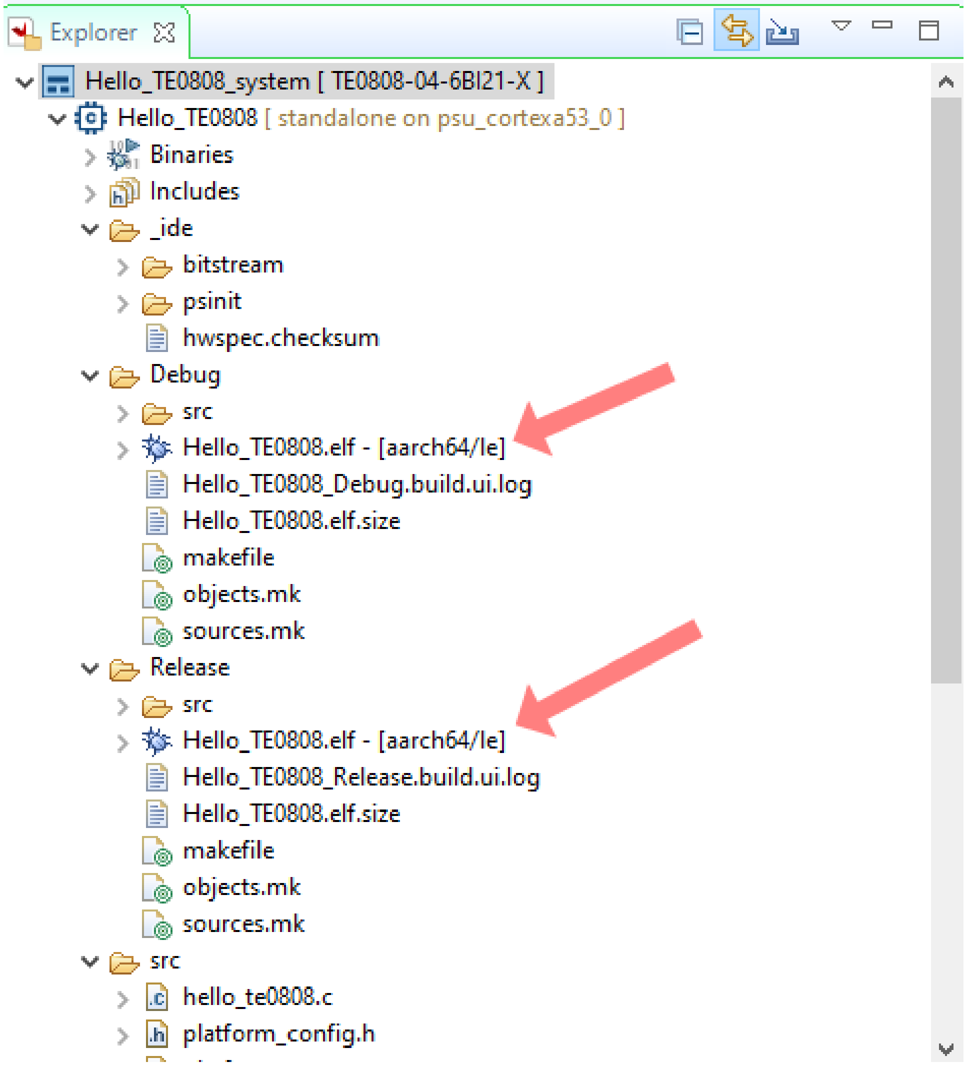
1. Select the project in the Explore Window and change active configuration (Debug or Release Build) and build design



2. To build selected configuration again, use build symbol or right click on the application project in the Explore Window and press "**Build Project**" or "**Clean Project**" to generate/regenerate the application project elf files



3. elf file will be generated in the corresponding release/debug folder



```
Hello_TE0808_system [ TE0808-04-6BI21-X ]
  Hello_TE0808 [ standalone on psu_cortexa53_0 ]
    Binaries
    Includes
    _ide
      bitstream
      psinit
      hwspec.checksum
    Debug
      src
      Hello_TE0808.elf - [aarch64/le]
      Hello_TE0808_Debug.build.ui.log
      Hello_TE0808.elf.size
      makefile
      objects.mk
      sources.mk
    Release
      src
      Hello_TE0808.elf - [aarch64/le]
      Hello_TE0808_Release.build.ui.log
      Hello_TE0808.elf.size
      makefile
      objects.mk
      sources.mk
    src
      hello_te0808.c
      platform_config.h
```

Debug Software Example

working in process .... coming soon

Create Boot Files

working in process .... coming soon

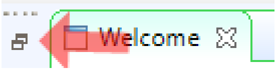
Zynq

ZynqMP

Microblaze

Linux Software Application

Linux application need a linux domain, which must be generated at first (see CreateDomain). Reference Designs, which includes linux design, generate a linux domain after Vitis project generation with vivado. After implementation/bitstream generation run on Vivado TCL console: **TE::sw_run_vitis**

- **Scripts generate Platform and application which are defined in <reference design base folder>\sw_lib\apps-list.csv. The overview window will not closed by default in this case, please close or minimise this vitis tab(****).**

For manual linux project generation, see subsections below.

working in process .... coming soon

Create Application

working in process .... coming soon

Build Application

working in process .... coming soon

Debug Software Example

working in process .... coming soon

Create Boot Files

working in process .... coming soon

Zynq

ZynqMP

Microblaze

References

- https://www.xilinx.com/products/design-tools/vitis/vitis-platform.html
- https://www.xilinx.com/html_docs/xilinx2019_2/vitis_doc/index.html
    - Note Replace <xilinx2019_2> of the URL with your Version (>= 19_2)