

Vitis

Table of content

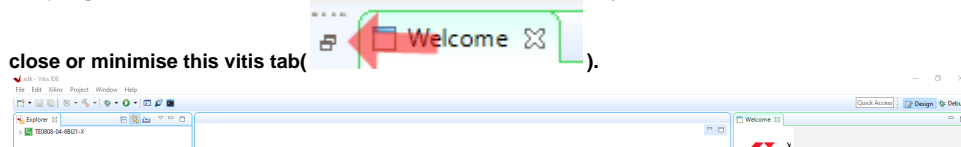
- [Table of content](#)
- [Create Vitis Project from Vivado](#)
 - [Create XSA with Vivado](#)
 - [Include local repositories](#)
 - [Create Platform project for XSA](#)
 - [Create Domain](#)
 - [Modify BSP Settings of the different Domains](#)
 - [Update Platform with modified XSA](#)
 - [Build Platform Project](#)
- [Bare-metal Software Application](#)
 - [Create Application](#)
 - [Build Application](#)
 - [Debug Software Example](#)
 - [Create Boot Files](#)
 - [Zynq](#)
 - [ZynqMP](#)
 - [Microblaze](#)
- [Linux Software Application](#)
 - [Create Application](#)
 - [Build Application](#)
 - [Debug Software Example](#)
 - [Create Boot Files](#)
 - [Zynq](#)
 - [ZynqMP](#)
 - [Microblaze](#)
- [References](#)

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**

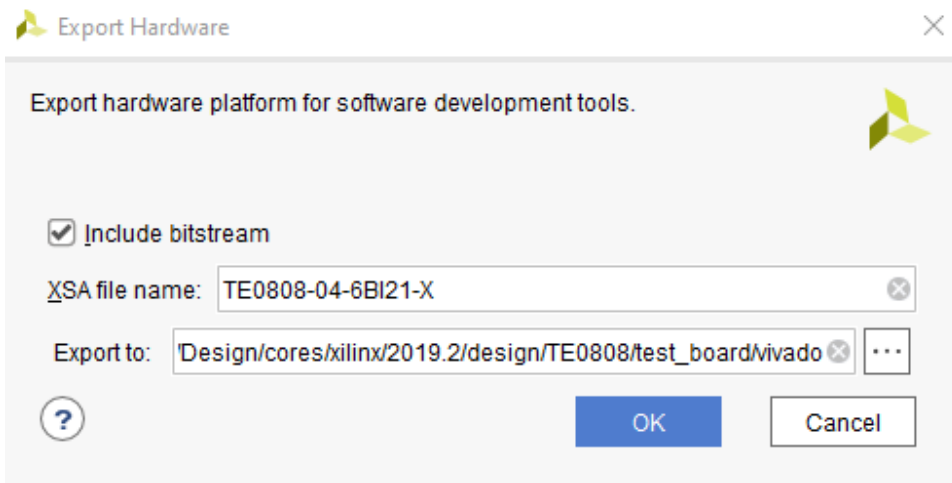


To create all manually see subsections below.

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.

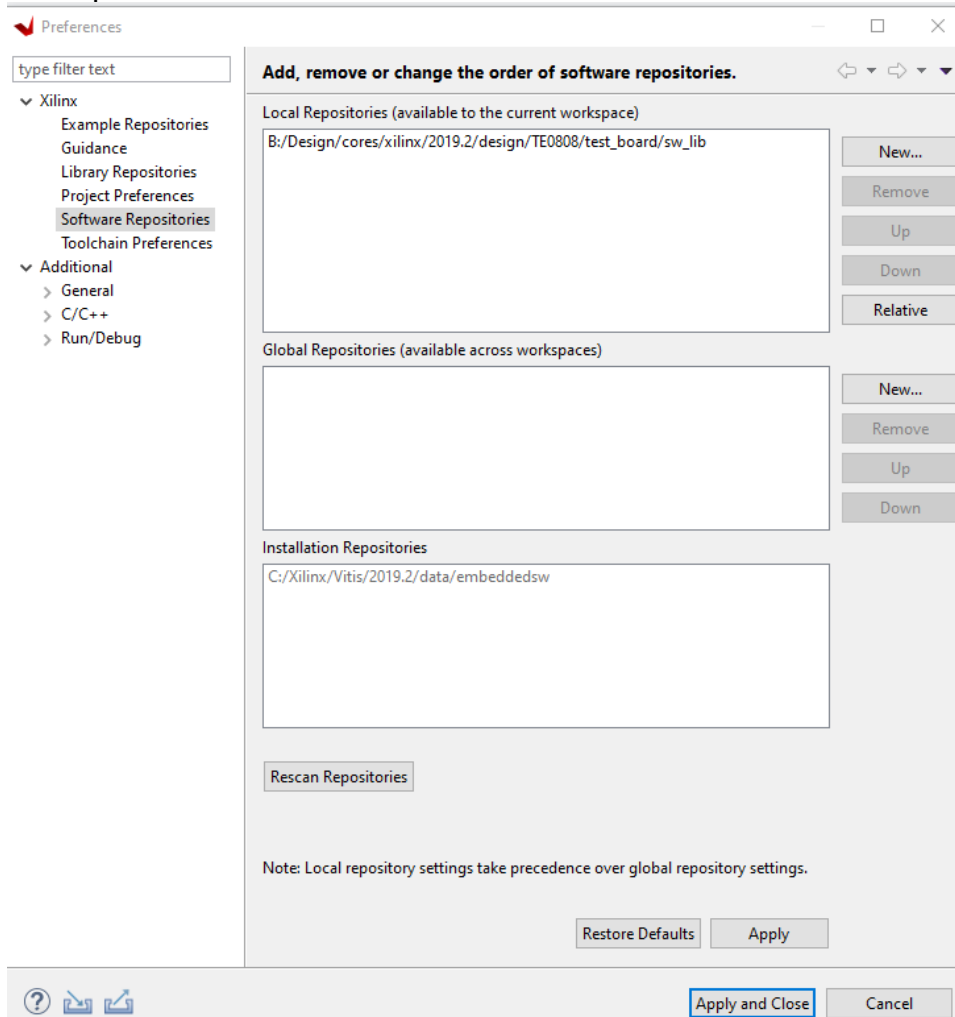
- **File Export Hardware...**



Include local repositories

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

- **Xilinx Repositories**



Create Platform project for XSA

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

2. Add name

New Platform Project

Create new platform project

Enter a name for your platform project

Project name:

☒ Use default location

Location:

Choose file system:

System Project 1

Baremetal App

A72_0 Baremetal Domain

System Project 2

Linux App 1

Linux App 2

AI Engine App

A72 Linux Domain

AI Engine Domain

Platform

- 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

New Platform Project

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.

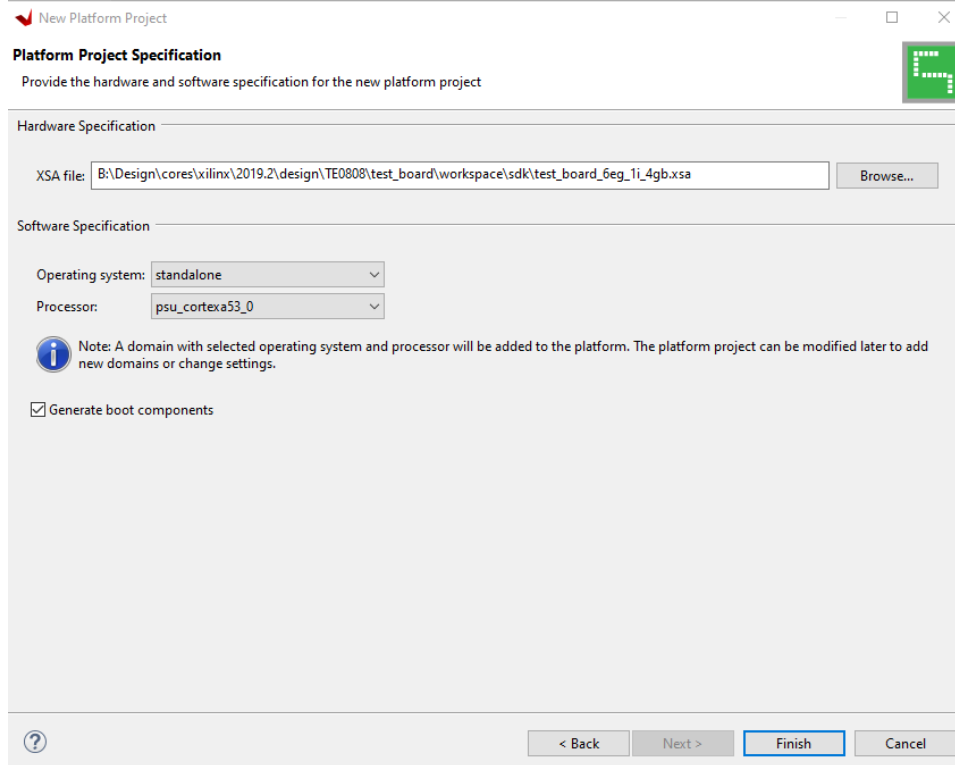
☐ Create from existing platform

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

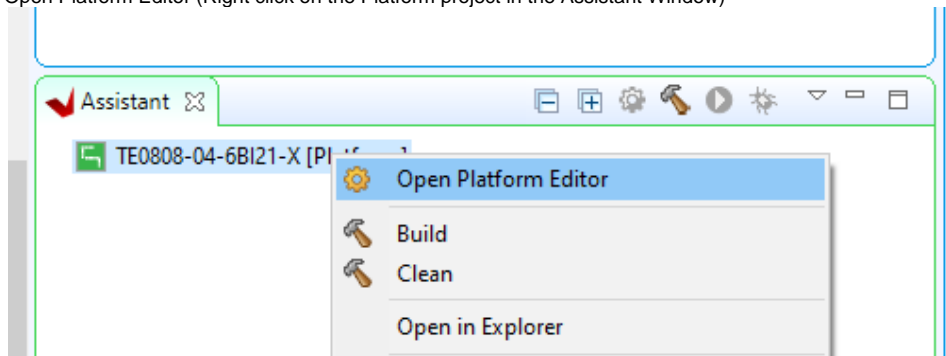


Create Domain

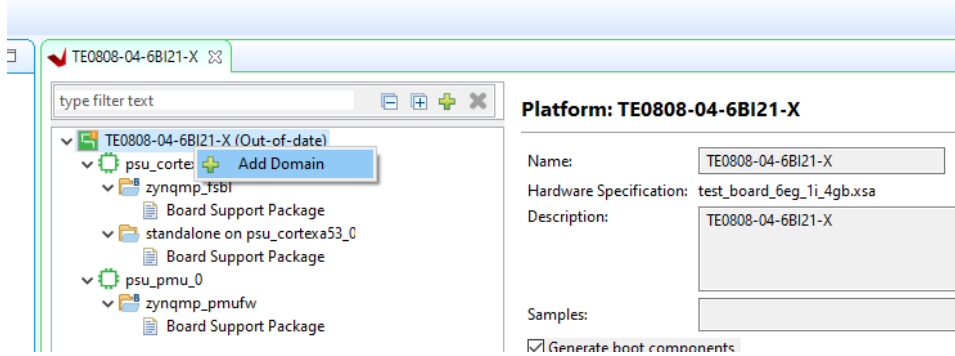
- Following Domains will be generated with Platform Project generation (if selected):
 - Stand-alone Domain for Stand-alone application (default selected on platform generation step)
 - FSBL Domain: for Zynq and ZynqMP devices
 - PMU Domain: for ZynqMP devices

Add new domain:

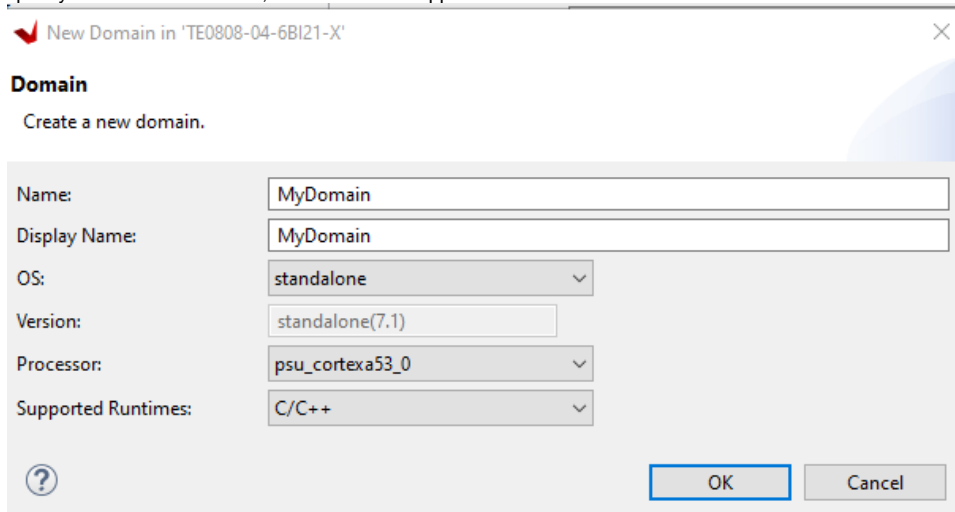
1. Open Platform Editor (Right click on the Platform project in the Assistant Window)



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

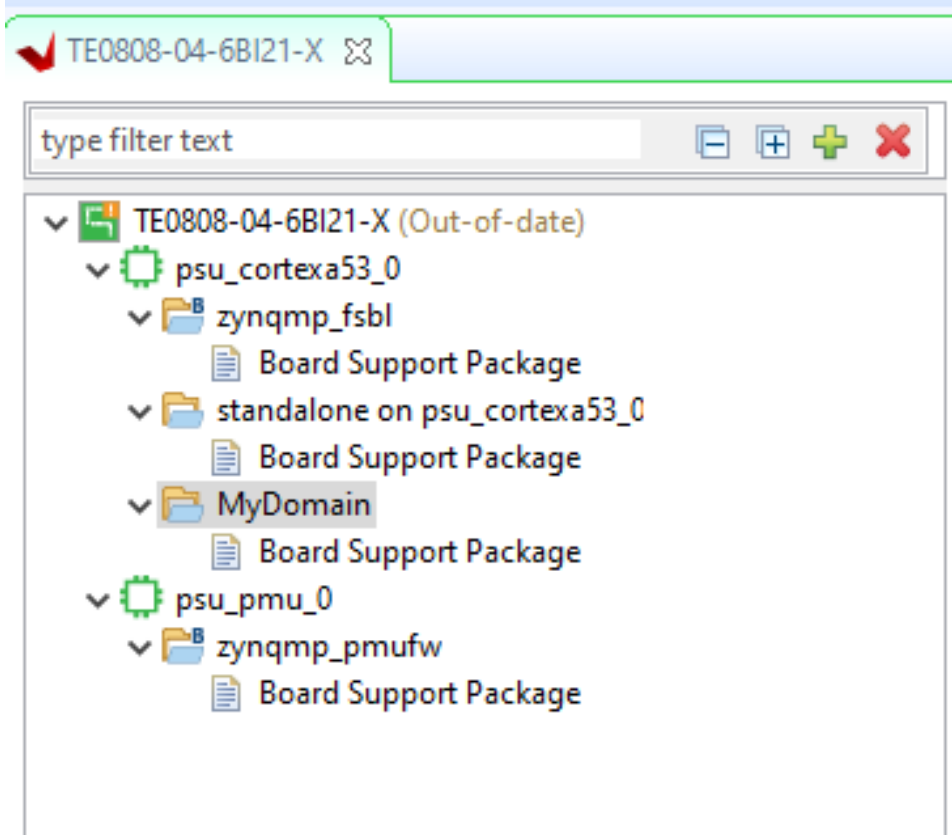


3. Specify Name and Select OS, Processor and supported runtime:



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.)

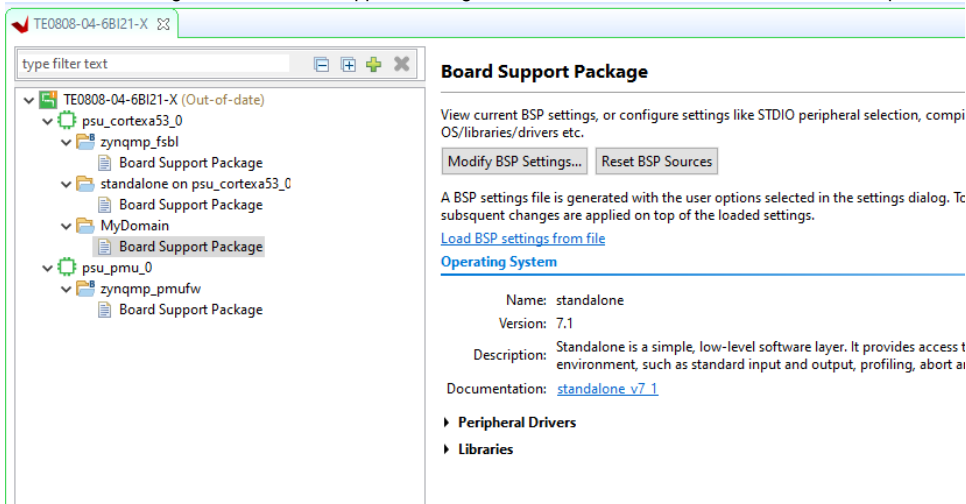
4. Domain will be generated and can be selected for applications:



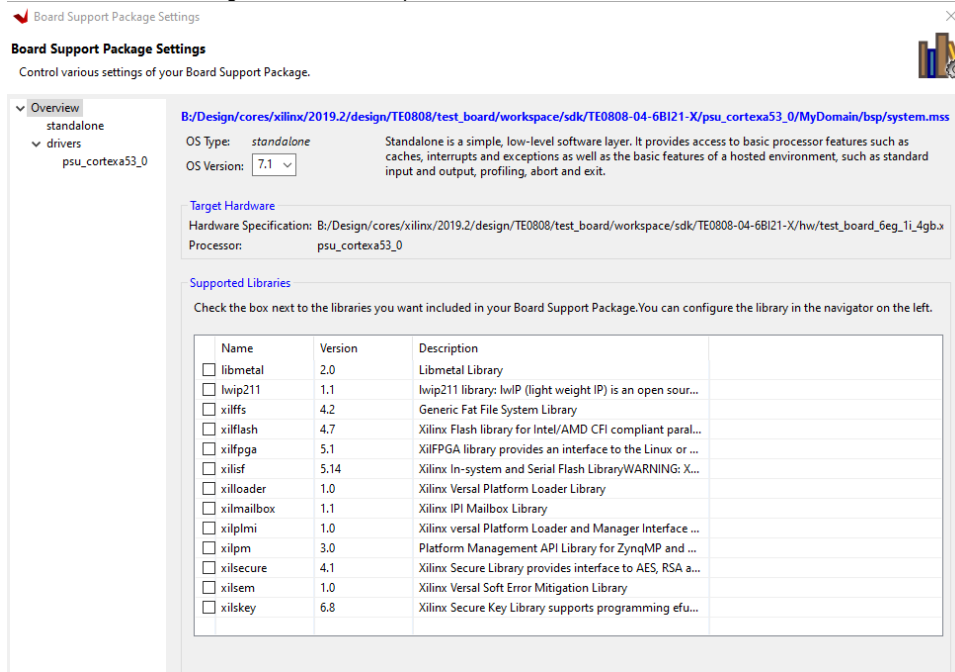
Modify BSP Settings of the different Domains

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..."

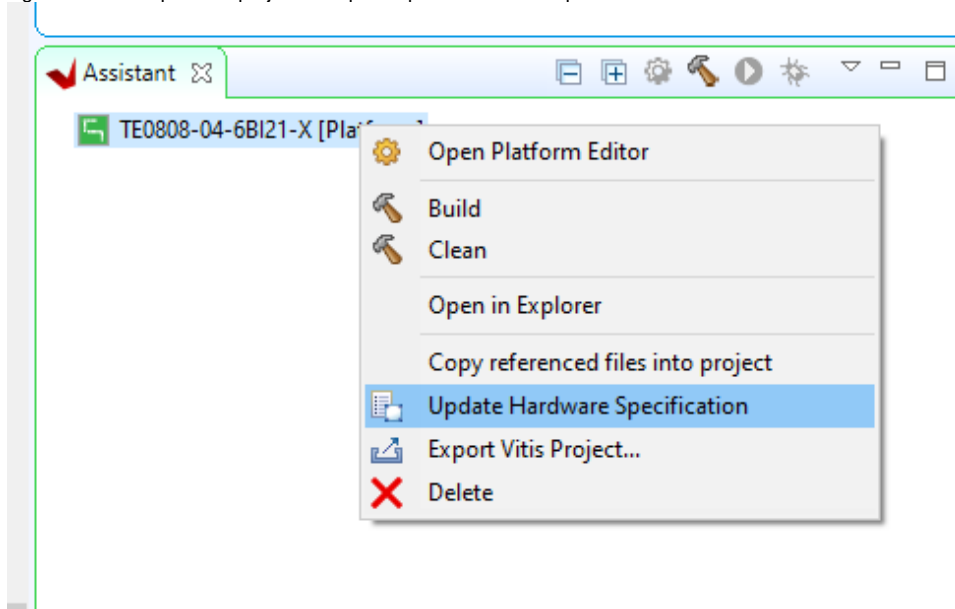


2. Add need drivers or change default BSP setup



Update Platform with modified XSA

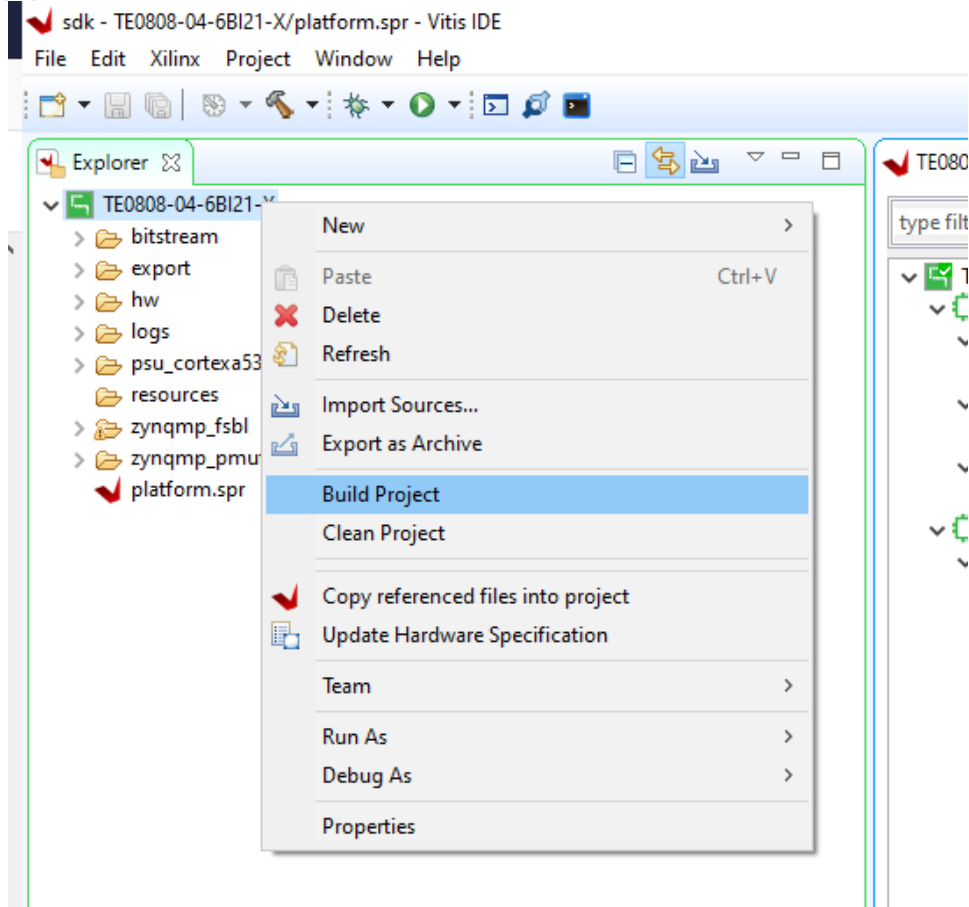
1. Right click on the platform project and pes "Update hardware Specification"



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

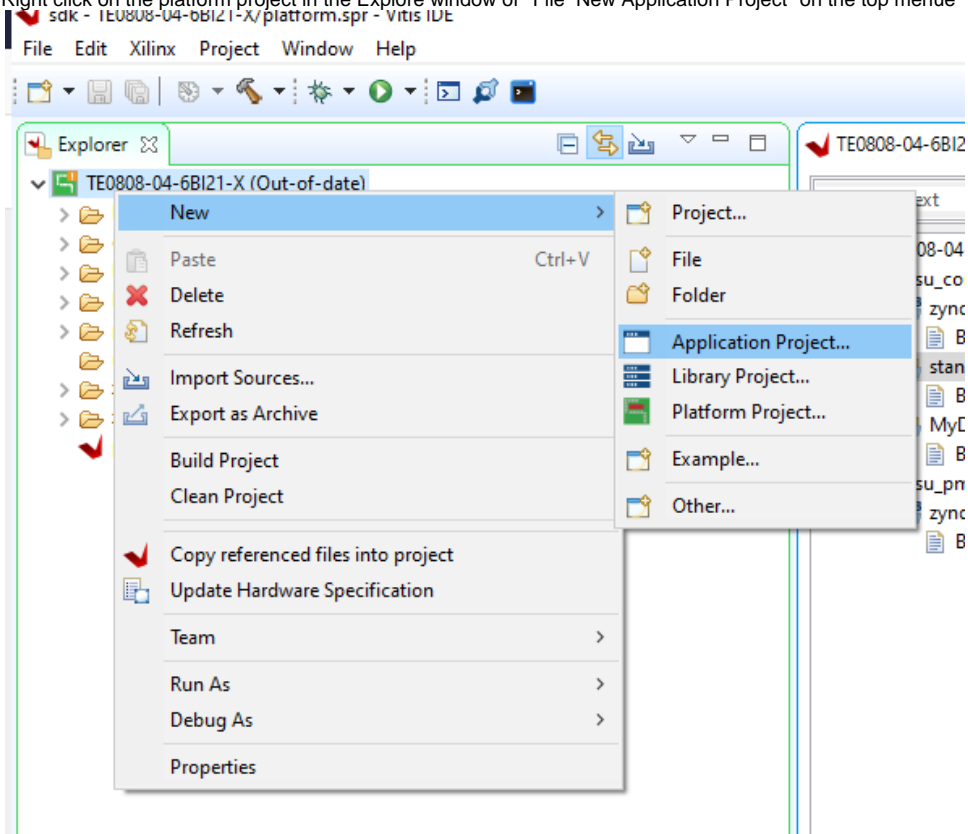
close or minimise this vitis tab().

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.

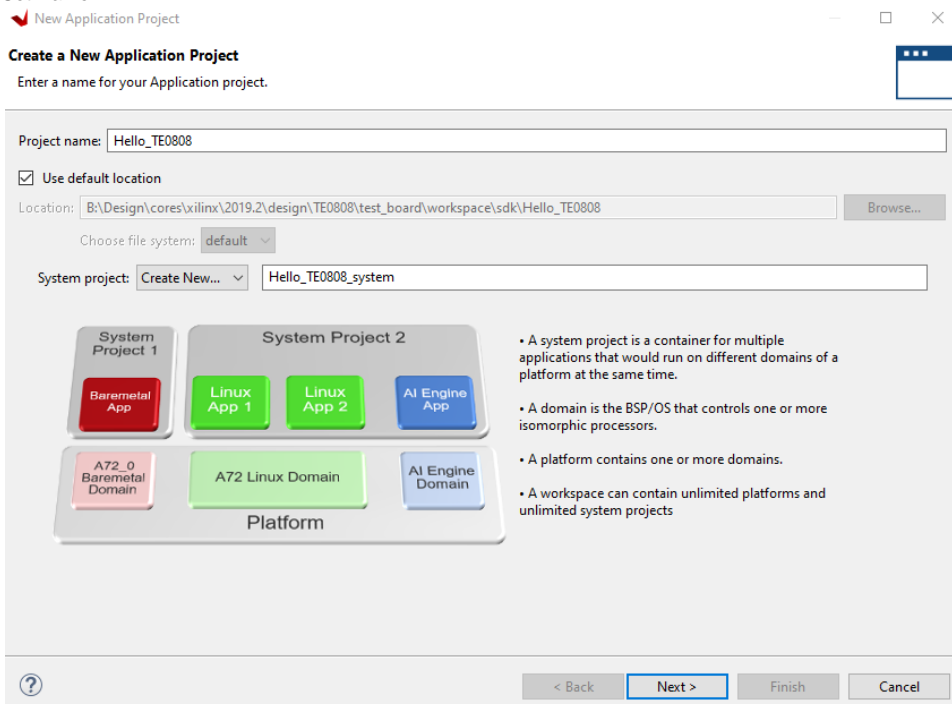
For manual bare metal project generation, see subsections below.

Create Application

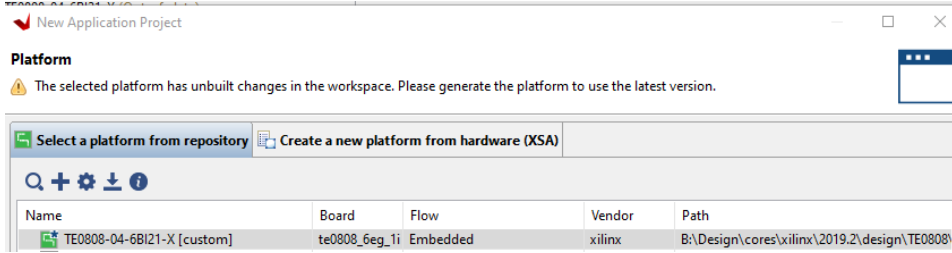
1. Right click on the platform project in the Explorer window or "File New Application Project" on the top menu



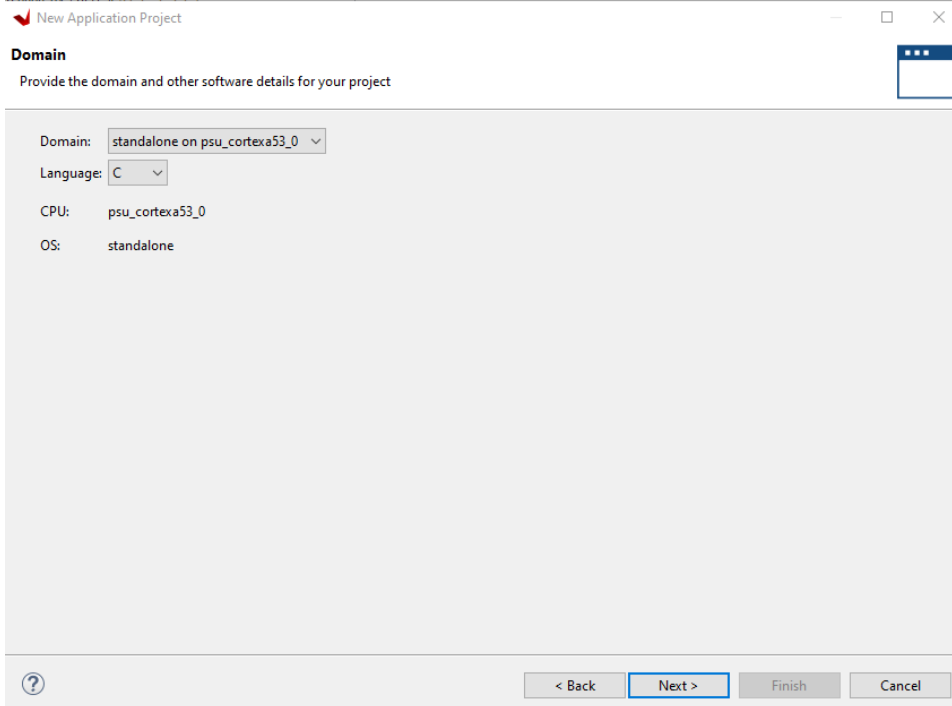
2. Set Name:



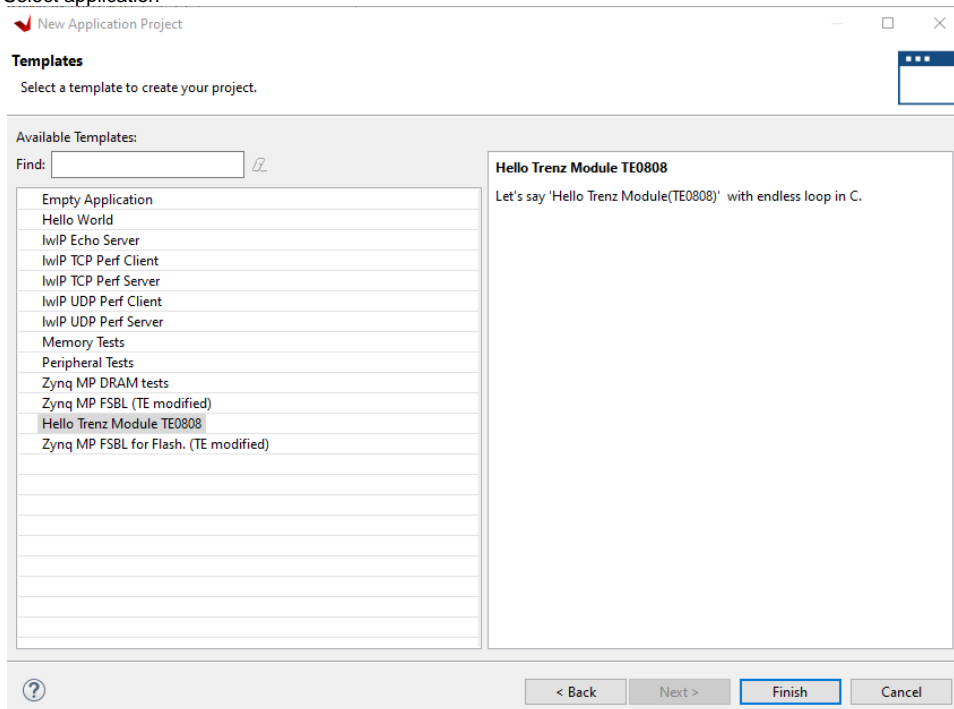
3. Select Platform



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



5. Select application

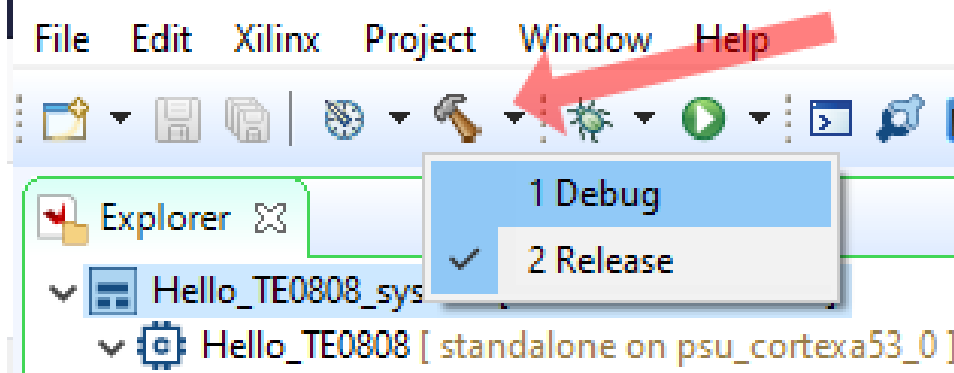


Note: In case application needs some Xilinx libraries (shown as note, in case the app can't be selected for generation), add these libraries to the domain BSP before you try to generate the application.

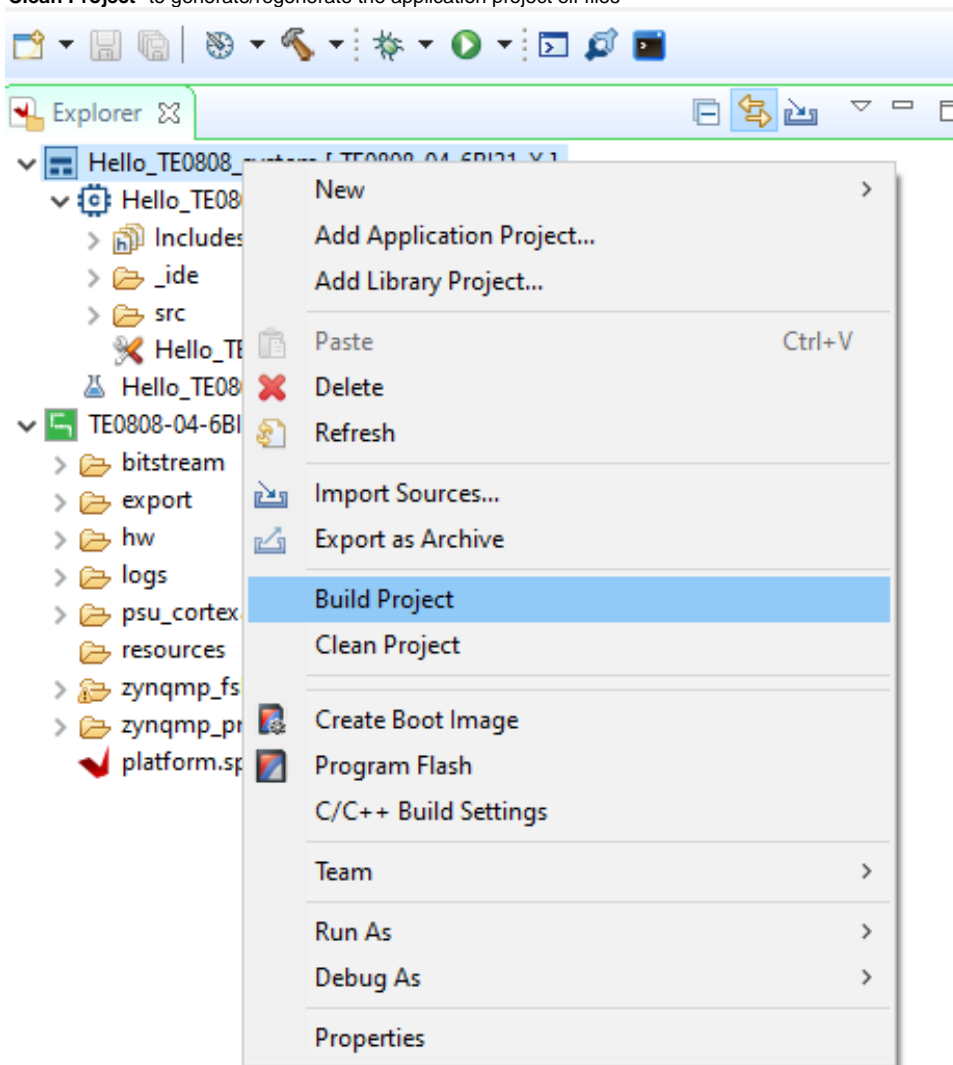
Build Application

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

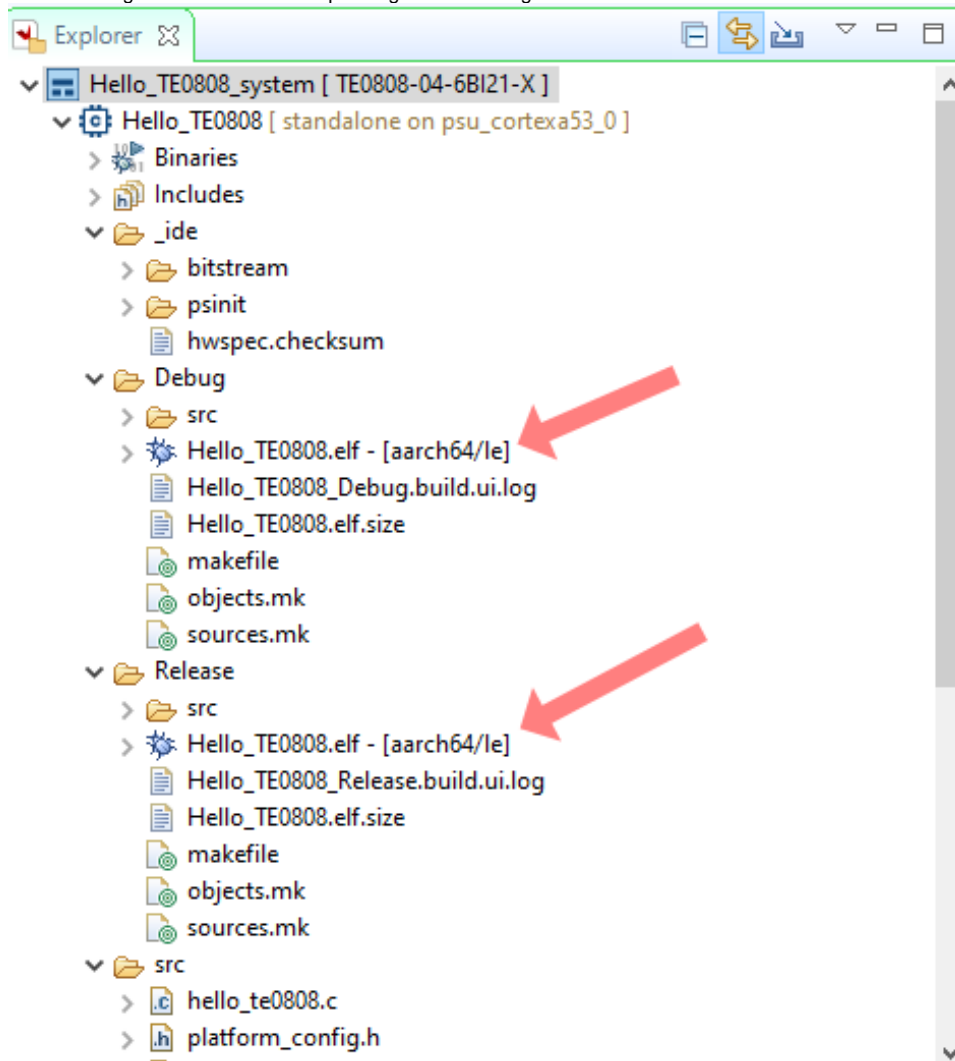
1. Select the project in the Explore Window and change active configuration (Debug or Release Build) and build design



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



Debug Software Example

working in process coming soon

Create Boot Files

working in process coming soon

Zynq

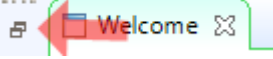
ZynqMP

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)