PetaLinux KICKstart

PetaLinux is brand name used by AMD , it is based on Yocto and pretty decent mainstream kernel, what Petalinux adds is the HSI (Hardware Software Interface from Vivado) and special tools for boot image creation.

Table of Content

Petalinux Project Creation- Short HOWTO

Getting Linux working on Zynq is very simple, following steps are required

1. Vivado/SDK/PetaLinux 20xx.x installed (important do not mix versions!)
2. Create Vivado Project, configure PS, Export HDF (XSA for 19.2 and newer)
3. Ceate new PetaLinux project
4. Import HDF(XSA for 19.2 and newer) into project
5. petalinux-build
6. copy boot.bin and image.ub to SD Card (only 2 files no more)

This is generic how-to, everything is setup for your by the Vivado->PetaLinux flow. Note, the boot.bin generated by PetaLinux may not always work, in such case it is recommended to make the boot.bin with SDK-GUI or command line tools manually.

There is no need to install anything else, or to fetch anything from any github repos, etc.

PetaLinux 2023.2

Petalinux 2023.2 is under review documentation for 2023.2 can be changed permanently at the moment

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - Petalinux is evaluated on Windows WSL with Ubuntu, see: AMD Tools and Win10 WSL
    - other VMs or native Linux Distribution is possible, but maybe changes are needed.
- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vitis (SDK+Vivado) 2023.2 with PetaLinux 2023.2
  - Note Petalinux is also available from Vivado/Vitis Installation setup
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version

1. Use bash as terminal:
   a. $ **sudo dpkg-reconfigure dash**
      i. press no
2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
   a. Install packages(see https://support.xilinx.com/s/article/000035572?language=en_US PetaLinux_2023.2_OS_Package_List.xlsx)
      i. Note:
         1. From excel additional to python3 also python was recommended which is not longer possible to install
         2. Additionally bc and was libtinfo5 added
         3. Add subversion (need only in case svn is used)
         4. Add u-boot-tools (need to generate own boot.scr file instead of petalinux version)

ii. **sudo apt-get install iproute2 gawk python3 build-essential gcc git make net-tools libncurses5-dev tftpd zlib1g-dev libssl-dev flex bison libselinux1 gnupg wget git diffstat chrpath socat xterm autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib automake zlib1g:i386 screen pax gzip cpio python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint bc libtinfo5 subversion u-boot-tools -y**

3. Install petalinux (Note: do not start from shared folder, copy installer into home directory)
    a. $ **sudo mkdir /tools**
    b. $ **sudo mkdir /tools/Xilinx**
    c. $ **sudo chown <owner>:<owner> /tools/Xilinx**
    d. $ **copy FPGAs_AdaptiveSoCs_Unified_2023.2_1013_2256.tar.gz into ~/**
    e. $ **tar -xvf FPGAs_AdaptiveSoCs_Unified_2023.2_1013_2256.tar.gz**
    f. $ **cd FPGAs_AdaptiveSoCs_Unified_2023.2_1013_2256**
    g. $ **./xsetup**
4. source enviroment
    a. $ **source /tools/Xilinx/PetaLinux/2023.2/tool/settings.sh**
5. Deactivate Webtalk:
    a. $ **petalinux-util --webtalk off**

- Note:
    - There is no need to install anything else, or to fetch anything from any github repos, etc.
    - It is recommended to test the installation by creating a dummy template project and building it.
        - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
        - Install:$ **petalinux-create -t project -s <path-to-bsp>**
        - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step 2.

Basic Note to TE Petalinux Templates, see: PetaLinux TE-Template#PetaLinux2023.2

New with 2023.2 and newer:

- ... TBD

New with 2022.2 and newer:

- Separate Device Tree for U-boot possible
- Device tree for uboot must be included into the Boot.bin

New with 2020.2 and newer:

- Trenz FSBL patches are available for petalinux now (beta, vitis template recommended at the moment)
- Xilinx changes:
    - **boot.src** is need for uboot now (will be generated with petalinux) see:
        - Distro Boot with Boot.scr
        - xilinx-wiki.atlassian "Using Distro Boot With Xilinx U-Boot"
    - "**petalinux-devtool**" command is need to update user layer with uboot and kernel changes

1. PetaLinux Working Environment (see UG1144)
    - PetaLinux Tools Installation is completed.
    - (Note: do not create project from shared folder, use home directory)
    - "/bin/sh" is bash
    - Set Working Environment:
        - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. (optional to create project from scretch instead of Trenz Templates) Create a New Project (see UG1144):
    a. $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**

        - <CPU_TYPE>: zynqMP, zynq, microblaze
        - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
    - Export Hardware Definition File (*.xsa) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
        - change to PetaLinux root folder:
        - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-config**
    - Run:$ **petalinux-config -c u-boot**
    - Run:$ **petalinux-config -c kernel**
    - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-build**
6. Generate Boot.bin with Vitis tools (recommended)
    a. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" for BOOT.BIN generation, also boot.scr is used (put separate on SD or include into Boot.bin for QSPI boot only). It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with Vitis tools.
7. Generate Boot.bin with Petalinux
    a. Run:$ **petalinux-package --boot <command options>**
    b. https://docs.xilinx.com/r/en-US/ug1144-petalinux-tools-reference-guide/petalinux-package-boot

Petalinux Configuration

Use local sstate cache and downlods

- Download (https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html):
    - aarch64 sstate-cache
        - for Versal, U+Zynq, RFSoC projects
    - arm sstate-cache
        - for 7 series Zynq projects
    - microblaze sstate-cache
        - for microblaze full projects
    - Downloads
        - forall types
- extract files for example in
    - mkdir ~/design/sstate-cache
    - cd ~/design/sstate-cache
    - copy archives into this folder
    - tar -xfv sstate_aarch64_2023.2_10121051.tar.gz
    - tar -xfv sstate_arm_2023.2_10121051.tar.gz
    - tar -xfv sstate_microblaze_2023.2_10121051.tar.gz
    - mkdir ~/design/sstate-cache/downloads_2023.2
    - cd ~/design/sstate-cache/downloads_2023.2
    - copy archives into this folder
    - tar -xfv downloads_2023.2_10121051.tar.gz
    - delete archive files
- change to the petalinux project
    - **petalinux-config**
        - for sstate-cache
            - -> Yocto Settings->Local sstate feeds settings->local sstate feeds url
                - ~/design/sstate-cache/sstate_aarch64_2023.2/aarch64
                - ~/design/sstate-cache/sstate_aarch64_2023.2/arm
                - ~/design/sstate-cache/sstate_aarch64_2023.2/mb-full
                - original: http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX_MAJOR_VER}/aarch64/sstate-cache
        - for downloads
            - -> Yocto Settings-> Add pre-mirror url
                - file://~/design/sstate-cache/downloads_2023.2/downloads
                - orginal:http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX_MAJOR_VER}/downloads
- Clear petalinux project
    - **petalinux-build -x mrproper**

Configuration menues and files

Most settings can be changed with menu-config:

- $ **petalinux-config**
- $ **petalinux-config -c u-boot**
- $ **petalinux-config -c kernel**
- $ **petalinux-config -c rootfs**

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

CONFIG	• project-spec/config /config	• changes with "**petalinux-config**" will be saved here
U-Boot	• recipes-bsp/u-boot /files • recipes-bsp/u-boot /files/platform-top.h • recipes-bsp/uboot-device-tree/files /ystem-user.dtsi	• changes with "**petalinux-config -c u-boot**" will be add to meta-user as recipes • optional overwrite, add UBoot settings on platform-top.h • Important:: Separate Device tree is disabled on default project generation, it must be enabled on **petalinux-config** menue under u-boot configuration otherwhise linux device tree will be used
Device Tree	• recipes-bsp/device-tree/files/system-user.dtsi • recipes-bsp/device-tree/files/pl-custom.dtsi	• overwrite or add device tree attributes
Kernel	• recipes-kernel/linux /linux-xlnx/	• changes with "**petalinux-config -c kernel**" will be add to meta-user as recipes
Apps	• project-spec/meta-user/recipes-apps	• add simple new app with "**petalinux-create -t apps -n myapp --enable**" • enable/disable with "**petalinux-config -c rootfs**"
ROOTFS	• project-spec/config /rootfs_config • project-spec/meta-user/conf/user-rootfsconfig	• "**petalinux-config -c rootfs**" will save changes in the general config spec, only apps will be saved in user-rootfsconfig also • user-rootfsconfig must be changed manually and will read after rootfs_config is used
Xilinx generated configuration	• project-spec/config/	Note: config and rootfs_config are shared at the moment, they include user and xilinx default changes from XSA import

Additional Descriptions

- Debian Installation
- Distro Boot with Boot.scr

- How to install the linux-rt (Real-Time) patch
- Petalinux General Hints and Note
- PetaLinux TE-Template
- Petalinux Troubleshoot
- Reference Designs with PetaLinux

References

1. PetaLinux Tools Documentation - Reference Guide (UG1144)
2. PetaLinux Tools Documentation - PetaLinux Command Line Reference (UG1157)
3. https://www.devicetree.org/
    a. https://www.devicetree.org/specifications/
4. https://github.com/Xilinx/linux-xlnx/tree/master/Documentation/devicetree/bindings

---

PetaLinux 2022.2

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
    - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
        - with OracleVM:
            - VM Setup:
                - RAM: >= 8GB
                - CPU: >= 4
                - HDD: 200GB dynanically
                    - ubuntu-20.04-desktop-amd64.iso
                    - install vm guest additions
                - Network: network bridge
                - optional: add shared folder, enable drag and drop

> **Alternative to Oracle VM**
>
> AMD Tools and Win10 WSL

- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
    - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
        - Example: Use Vitis (SDK+Vivado) 2021.2 with PetaLinux 2021.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
    1. Use bash as terminal:
        a. $ **sudo dpkg-reconfigure dash**
            i. press no
    2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries

a. https://www.xilinx.com/support/answers/73296.html

i. download plnx-env-setup.sh and run

b. $ **sudo apt-get update**

c. $ **sudo apt-get install iproute2 gawk python3 python build-essential gcc git make net-tools libncurses5-dev tftpd zlib1g-dev libssl-dev flex bison libselinux1 gnupg wget git-core diffstat chrpath socat xterm autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib automake zlib1g:i386 screen pax gzip cpio python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3 -y**

i. **This is necessary because lnx-env-setup.sh seems to be not install all packages correctly**

3. Install petalinux (Note: do not start from shared folder, copy installer into home directory)

a. $ **mkdir -p ~/petalinux/2022.2**

b. **copy petalinux-v2022.2-final-installer.run into ~/petalinux/2022.2**

c. $ ./petalinux-v2020.2-final-installer.run

4. source enviroment

a. $ **source ~/petalinux/2022.2/settings.sh**

5. Deactivate Webtalk:

a. $ **petalinux-util --webtalk off**

- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install:$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step 2.

Basic Note to TE Petalinux Templates, see: PetaLinux TE-Template#PetaLinux2021.2

New with 2022.2 and newer:

- Separate Device Tree for U-boot possible
- Device tree for uboot must be included into the Boot.bin
- ... TBD

New with 2020.2 and newer:

- Trenz FSBL patches are available for petalinux now (beta, vitis template recommended at the moment)
- Xilinx changes:
  - **boot.src** is need for uboot now (will be generated with petalinux) see:
    - Distro Boot with Boot.scr
    - xilinx-wiki.atlassian "Using Distro Boot With Xilinx U-Boot"
  - "**petalinux-devtool**" command is need to update user layer with uboot and kernel changes

1. PetaLinux Working Environment (see UG1144)
   - PetaLinux Tools Installation is completed.
   - (Note: do not create project from shared folder, use home directory)
   - "/bin/sh" is bash
   - Set Working Environment:
     - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. (optional to create project from scretch instead of Trenz Templates) Create a New Project (see UG1144):
   a. $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**

- <CPU_TYPE>: zynqMP, zynq, microblaze
- <PROJECT_NAME>:The name of the project you are building

3. Import Hardware Configuration (see UG1144):
    - Export Hardware Definition File (*.xsa) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
        - change to PetaLinux root folder:
        - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-config**
    - Run:$ **petalinux-config -c u-boot**
    - Run:$ **petalinux-config -c kernel**
    - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-build**
6. Generate Boot.bin with Vitis tools (recommended)
    a. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" for BOOT.BIN generation, also boot.scr is used (put separate on SD or include into Boot.bin for QSPI boot only). It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with Vitis tools.
7. Generate Boot.bin with Petalinux
    a. Run:$ **petalinux-package --boot <command options>**
    b. https://docs.xilinx.com/r/en-US/ug1144-petalinux-tools-reference-guide/petalinux-package-boot

Petalinux Configuration

Use local sstate cache and downlods

- Download (https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html):
    - aarch64 sstate-cache
        - for Versal, U+Zynq, RFSoC projects
    - arm sstate-cache
        - for 7 series Zynq projects
    - mbfull sstate-cache
        - for microblaze full projects
    - mblite sstate-cache
        - for microblaze lite projects
    - Downloads
        - forall types
- extract files for example in
    - mkdir ~/design/sstate-cache
    - cd ~/design/sstate-cache
    - copy archives into this folder
    - tar -xfv sstate_aarch64_2022.2_10071807.tar.gz
    - tar -xfv sstate_arm_2022.2_10071807.tar.gz
    - tar -xfv sstate_microblaze_2022.2_10071807.tar.gz
    - mkdir ~/design/sstate-cache/downloads_2022.2
    - cd ~/design/sstate-cache/downloads_2022.2
    - copy archives into this folder
    - tar -xfv downloads_2022.2_10071807.tar.gz
    - delete archive files
- change to the petalinux project
    - **petalinux-config**
        - for sstate-cache
            - -> Yocto Settings->Local sstate feeds settings->local sstate feeds url
                - ~/design/sstate-cache/sstate_aarch64_20222/aarch64
                - ~/design/sstate-cache/sstate_aarch64_2022.2/arm
                - ~/design/sstate-cache/sstate_aarch64_2022.2/mb-full
                - original: http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX_MAJOR_VER}/aarch64/sstate-cache
        - for downloads
            - -> Yocto Settings-> Add pre-mirror url
                - file://~/design/sstate-cache/downloads_2021.2/downloads
                - orginal:http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX_MAJOR_VER}/downloads
- Clear petalinux project
    - **petalinux-build -x mrproper**

Configuration menues and files

Most settings can be changed with menu-config:

- $ **petalinux-config**
- $ **petalinux-config -c u-boot**
- $ **petalinux-config -c kernel**
- $ **petalinux-config -c rootfs**

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

CONFIG	• project-spec/config /config	• changes with "**petalinux-config**" will be saved here
U-Boot	• recipes-bsp/u-boot /files • recipes-bsp/u-boot /files/platform-top.h • recipes-bsp/uboot-device-tree/files /ystem-user.dtsi	• changes with "**petalinux-config -c u-boot**" will be add to meta-user as recipes • optional overwrite, add UBoot settings on platform-top.h • Important:: Separate Device tree is disabled on default project generation, it must be enabled on **petalinux-config** menue under u-boot configuration otherwhise linux device tree will be used
Device Tree	• recipes-bsp/device-tree/files/system-user.dtsi • recipes-bsp/device-tree/files/pl-custom.dtsi	• overwrite or add device tree attributes
Kernel	• recipes-kernel/linux /linux-xlnx/	• changes with "**petalinux-config -c kernel**" will be add to meta-user as recipes
Apps	• project-spec/meta-user/recipes-apps	• add simple new app with "**petalinux-create -t apps -n myapp --enable**" • enable/disable with "**petalinux-config -c rootfs**"
ROOTFS	• project-spec/config /rootfs_config • project-spec/meta-user/conf/user-rootfsconfig	• "**petalinux-config -c rootfs**" will save changes in the general config spec, only apps will be saved in user-rootfsconfig also • user-rootfsconfig must be changed manually and will read after rootfs_config is used
Xilinx generated configuration	• project-spec/config/	Note: config and rootfs_config are shared at the moment, they include user and xilinx default changes from XSA import

PetaLinux 2021.2

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
    - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
        - with OracleVM:
            - VM Setup:
                - RAM: >= 8GB
                - CPU: >= 4
                - HDD: 200GB dynanically
                    - ubuntu-20.04-desktop-amd64.iso
                    - install vm guest additions
                - Network: network bridge
                - optional: add shared folder, enable drag and drop

> **Alternative to Oracle VM**
>
> AMD Tools and Win10 WSL

- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
    - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
        - Example: Use Vitis (SDK+Vivado) 2021.2 with PetaLinux 2021.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
    1. Use bash as terminal:
        a. $ **sudo dpkg-reconfigure dash**
            i. press no
    2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
        a. https://www.xilinx.com/support/answers/73296.html
            i. download plnx-env-setup.sh and run
        b. $ **sudo apt-get update**
        c. $ **sudo apt-get install iproute2 gawk python3 python build-essential gcc git make net-tools libncurses5-dev tftpd zlib1g-dev libssl-dev flex bison libselinux1 gnupg wget git-core diffstat chrpath socat xterm autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib automake zlib1g:i386 screen pax gzip cpio python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3 -y**
            i. **This is necessary because lnx-env-setup.sh seems to be not install all packages correctly**
    3. Install petalinux (Note: do not start from shared folder, copy installer into home directory)
        a. $ **mkdir -p ~/petalinux/2021.2**
        b. **copy petalinux-v2021.2-final-installer.run into ~/petalinux/2021.2**
        c. $ ./petalinux-v2020.2-final-installer.run
    4. source enviroment
        a. $ **source ~/petalinux/2021.2/settings.sh**
    5. Deactivate Webtalk:
        a. $ **petalinux-util --webtalk off**
- Note:
    - There is no need to install anything else, or to fetch anything from any github repos, etc.
    - It is recommended to test the installation by creating a dummy template project and building it.
        - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
        - Install:$ **petalinux-create -t project -s <path-to-bsp>**
        - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step 2.

Basic Note to TE Petalinux Templates, see: PetaLinux TE-Template#PetaLinux2021.2

New with 2021.2 and newer:

- Separate Device Tree for U-boot possible
- Device tree for uboot must be included into the Boot.bin
- ... TBD

New with 2020.2 and newer:

- Trenz FSBL patches are available for petalinux now (beta, vitis template recommended at the moment)
- Xilinx changes:
  - **boot.src** is need for uboot now (will be generated with petalinux) see:
    - Distro Boot with Boot.scr
    - xilinx-wiki.atlassian "Using Distro Boot With Xilinx U-Boot"
  - "**petalinux-devtool**" command is need to update user layer with uboot and kernel changes

1. PetaLinux Working Environment (see UG1144)
   - PetaLinux Tools Installation is completed.
   - (Note: do not create project from shared folder, use home directory)
   - "/bin/sh" is bash
   - Set Working Environment:
     - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. (optional to create project from scretch instead of Trenz Templates) Create a New Project (see UG1144):
   a. $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**
      - <CPU_TYPE>: zynqMP, zynq, microblaze
      - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
   - Export Hardware Definition File (*.xsa) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
     - change to PetaLinux root folder:
     - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-config**
   - Run:$ **petalinux-config -c u-boot**
   - Run:$ **petalinux-config -c kernel**
   - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" for BOOT.BIN generation, also boot.scr is used (put separate on SD or include into Boot.bin for QSPI boot only). It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with Vitis tools.

Petalinux Configuration

Use local sstate cache and downlods

- Download (https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html):
  - aarch64 sstate-cache
    - for Versal, U+Zynq, RFSoC projects
  - arm sstate-cache

        - for 7 series Zynq projects
    - mbfull sstate-cache
        - for microblaze full projects
    - mblite sstate-cache
        - for microblaze lite projects
    - Downloads
        - forall types
- extract files for example in
    - mkdir ~/design/sstate-cache
    - cd ~/design/sstate-cache
    - copy archives into this folder
    - tar -xfv sstate_aarch64_2021.2.tar.gz
    - tar -xfv sstate_arm_2021.2.tar.gz
    - tar -xfv sstate_mb-full_2021.2.tar.gz
    - tar -xfv sstate_mb-lite_2021.2.tar.gz
    - mkdir ~/design/sstate-cache/downloads_2021.2
    - cd ~/design/sstate-cache/downloads_2021.2
    - copy archives into this folder
    - tar -xfv downloads_2021.2.tar.gz
    - delete archive files
- change to the petalinux project
    - **petalinux-config**
        - for sstate-cache
            - -> Yocto Settings->Local sstate feeds settings->local sstate feeds url
                - ~/design/sstate-cache/sstate_aarch64_2021.2/aarch64
                - ~/design/sstate-cache/sstate_aarch64_2021.2/arm
                - ~/design/sstate-cache/sstate_aarch64_2021.2/mb-full
                - ~/design/sstate-cache/sstate_aarch64_2021.2/mb-lite
                - original: http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX_MAJOR_VER}/aarch64/sstate-cache
        - for downloads
            - -> Yocto Settings-> Add pre-mirror url
                - file://~/design/sstate-cache/downloads_2021.2/downloads
                - orginal:http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX_MAJOR_VER}/downloads
- Clear petalinux project
    - **petalinux-build -x mrproper**

Configuration menues and files

Most settings can be changed with menu-config:

- $ **petalinux-config**
- $ **petalinux-config -c u-boot**
- $ **petalinux-config -c kernel**
- $ **petalinux-config -c rootfs**

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

CONFIG	• project-spec/config /config	• changes with "**petalinux-config**" will be saved here
U-Boot	• recipes-bsp/u-boot /files • recipes-bsp/u-boot /files/platform-top.h • recipes-bsp/uboot-device-tree/files /ystem-user.dtsi	• changes with "**petalinux-config -c u-boot**" will be add to meta-user as recipes • optional overwrite, add UBoot settings on platform-top.h • Important:: Separate Device tree is disabled on default project generation, it must be enabled on **petalinux-config** menue under u-boot configuration otherwhise linux device tree will be used

Device Tree	• recipes-bsp/device-tree/files/system-user.dtsi • recipes-bsp/device-tree/files/pl-custom.dtsi	• overwrite or add device tree attributes
Kernel	• recipes-kernel/linux/linux-xlnx/	• changes with "**petalinux-config -c kernel**" will be add to meta-user as recipes
Apps	• project-spec/meta-user/recipes-apps	• add simple new app with "**petalinux-create -t apps -n myapp --enable**" • enable/disable with "**petalinux-config -c rootfs**"
ROOTFS	• project-spec/config/rootfs_config • project-spec/meta-user/conf/user-rootfsconfig	• "**petalinux-config -c rootfs**" will save changes in the general config spec, only apps will be saved in user-rootfsconfig also • user-rootfsconfig must be changed manually and will read after rootfs_config is used
Xilinx generated configuration	• project-spec/config/	Note: config and rootfs_config are shared at the moment, they include user and xilinx default changes from XSA import

PetaLinux 2020.2

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
    - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
        - with OracleVM:
            - VM Setup:
                - RAM: >= 8GB
                - CPU: >= 4
                - HDD: 200GB dynanically
                    - ubuntu-18.04-desktop-amd64.iso
                    - install vm guest additions
                    - update to 18.04.5 was done
                - Network: network bridge
                - optional: add shared folder, enable drag and drop

> **Alternative to Oracle VM**
>
> AMD Tools and Win10 WSL

- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
    - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.

        - Example: Use Vitis (SDK+Vivado) 2021.2 with PetaLinux 2020.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version

    1. Use bash as terminal:

        a. $ **sudo dpkg-reconfigure dash**

            i. press no
    2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries

        a. https://www.xilinx.com/support/answers/73296.html

            i. download plnx-env-setup.sh and run
        b. $ **sudo apt-get update**
        c. $ **sudo apt-get install iproute2 gawk python3 python build-essential gcc git make net-tools libncurses5-dev tftpd zlib1g-dev libssl-dev flex bison libselinux1 gnupg wget git-core diffstat chrpath socat xterm autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib automake zlib1g:i386 screen pax gzip cpio python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3 -y**

            i. **This is necessary because lnx-env-setup.sh seems to be not install all packages correctly**
    3. Install petalinux (Note: do not start from shared folder, copy installer into home directory)

        a. $ **mkdir -p ~/petalinux/2020.2**
        b. **copy petalinux-v2020.2-final-installer.run into ~/petalinux/2020.2**
        c. $ ./petalinux-v2020.2-final-installer.run
    4. source enviroment

        a. $ **source ~/petalinux/2020.2/settings.sh**
    5. Deactivate Webtalk:

        a. $ **petalinux-util --webtalk off**
- Note:

    - There is no need to install anything else, or to fetch anything from any github repos, etc.
    - It is recommended to test the installation by creating a dummy template project and building it.

        - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
        - Install:$ **petalinux-create -t project -s <path-to-bsp>**
        - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step 2.

Basic Note to TE Petalinux Templates, see: PetaLinux TE-Template#PetaLinux2020.2

New with 2020.2:

- Trenz FSBL patches are available for petalinux now (beta, vitis template recommended at the moment)
- Xilinx changes:
    - **boot.src** is need for uboot now (will be generated with petalinux) see:
        - Distro Boot with Boot.scr
        - xilinx-wiki.atlassian "Using Distro Boot With Xilinx U-Boot"
    - "**petalinux-devtool**" command is need to update user layer with uboot and kernel changes

1. PetaLinux Working Environment (see UG1144)
    - PetaLinux Tools Installation is completed.
    - (Note: do not create project from shared folder, use home directory)
    - "/bin/sh" is bash
    - Set Working Environment:
        - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. (optional to create project from scretch instead of Trenz Templates) Create a New Project (see UG1144):

a. $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**

- <CPU_TYPE>: zynqMP, zynq, microblaze
- <PROJECT_NAME>:The name of the project you are building

3. Import Hardware Configuration (see UG1144):
   - Export Hardware Definition File (*.xsa) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
     - change to PetaLinux root folder:
     - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-config**
   - Run:$ **petalinux-config -c u-boot**
   - Run:$ **petalinux-config -c kernel**
   - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" for BOOT.BIN generation, also boot.scr is used (put separate on SD or include into Boot.bin for QSPI boot only). It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with Vitis tools.

Petalinux Configuration

Most settings can be changed with menu-config*:

- $ **petalinux-config**
- $ **petalinux-config -c u-boot**
- $ **petalinux-config -c kernel**
- $ **petalinux-config -c rootfs**

* with **2020.2** kernel and u-boot changes must be exported to the user-layer with petalinux-devtool command.

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

CONFIG	• project-spec/config/config	• changes with "**petalinux-config**" will be saved here
U-Boot	• recipes-bsp/u-boot/files • recipes-bsp/u-boot/files /platform-top.h	• changes with "**petalinux-config -c u-boot**" will be add in the yocto workspace • force changes to the user layer run "**petalinux-devtool finish u-boot-xlnx ${PWD}/project-spec/meta-user/ -f**" • optional overwrite, add UBoot settings on platform-top.h
Device Tree	• recipes-bsp/device-tree/files /system-user.dtsi • recipes-bsp/device-tree/files /pl-custom.dtsi	• overwrite or add device tree attributes
Kernel	• recipes-kernel/linux/linux-xlnx/	• changes with "**petalinux-config -c kernel**" will be add in the yocto workspace • force changes to the user layer run "**petalinux-devtool finish linux-xlnx ${PWD} /project-spec/meta-user/ -f**"
Apps	• project-spec/meta-user /recipes-apps	• add simple new app with "**petalinux-create -t apps -n myapp --enable**" • enable/disable with "**petalinux-config -c rootfs**"

ROOTFS	• project-spec/config /rootfs_config • project-spec/meta-user/conf /user-rootfsconfig	• "**petalinux-config -c rootfs**" will save changes in the general config spec, only apps will be saved in user-rootfsconfig also • user-rootfsconfig must be changed manually and will read after rootfs_config is used
Xilinx generated configuration	• project-spec/config/	Note: config and rootfs_config are shared at the moment, they include user and xilinx default changes from XSA import

PetaLinux 2019.2

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
    - with OracleVM:
      - VM Setup:
        - RAM: >= 8GB
        - CPU: >= 4
        - HDD: 200GB dynanically
          - ubuntu-18.04-desktop-amd64.iso
          - install vm guest additions
        - Network: network bridge
        - optional: add shared folder, enable drag and drop
- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vitis (SDK+Vivado) 2019.2 with PetaLinux 2019.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Use bash as terminal:
     a. $ **sudo dpkg-reconfigure dash**
        i. press no
  2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
     a. $ **sudo apt-get update**
     b. $ **sudo apt-get install tofrodos iproute2 gawk make net-tools libncurses5-dev tftpd zlib1g:i386 libssl-dev flex bison libselinux1 gnupg wget diffstat chrpath socat xterm autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib build-essential screen pax gzip python 2.7.5 -y**
  3. Change owner of installation directory to non root
     a. $ **sudo chown** <owner>:<owner> **/opt/**
  4. Install petalinux (Note: do not start from shared folder, copy installer into home directory)
     a. $ **mkdir -p /opt/pkg/petalinux/2019.2**
     b. $ ./petalinux-v2019.2-final-installer.run /opt/pkg/petalinux/2019.2
  5. source enviroment
     a. $ **source /opt/pkg/petalinux/2019.2/settings.sh**
  6. Deactivate Webtalk:
     a. $ **petalinux-util --webtalk off**
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install:$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two. Basic Note to TE Petalinux Templates, see: PetaLinux TE-Template#Template-PetaLinux2019.2

1. PetaLinux Working Environment (see UG1144)
    - PetaLinux Tools Installation is completed.
    - (Note: do not create project from shared folder, use home directory)
    - "/bin/sh" is bash
    - Set Working Environment:
        - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. (optional to create project from scretch instead of Trenz Templates) Create a New Project (see UG1144):
    a. $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**

        - <CPU_TYPE>: zynqMP, zynq, microblaze
        - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
    - Export Hardware Definition File (*.xsa) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
        - change to PetaLinux root folder:
        - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-config**
    - Run:$ **petalinux-config -c u-boot**
    - Run:$ **petalinux-config -c kernel**
    - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" for BOOT.BIN generation. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with Vitis tools.

Petalinux Configuration

Most settings can be changed with menu-config:

- $ **petalinux-config**
- $ **petalinux-config -c u-boot**
- $ **petalinux-config -c kernel**
- $ **petalinux-config -c rootfs**

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	• recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	• recipes-bsp/device-tree/files/system-user.dtsi • recipes-bsp/device-tree/files/pl-custom.dtsi	overwrite, add device tree attributes
Kernel	• recipes-kernel/linux/linux-xlnx/	changes with "**petalinux-config -c kernel**" will be add here automatically
Apps	• project-spec/meta-user/recipes-apps	add simple new app with "**petalinux-create -t apps -n myapp --enable**" enable/disable with "**petalinux-config -c rootfs**"

PetaLinux 2018.3

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2018.3 with SDK 2018.3 and PetaLinux 2018.3
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Change owner of installation directory to non root (see Troubleshoot page)
  2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  3. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  4. Deactivate Webtalk: $ **petalinux-util --webtalk off**
  5. See also Petalinux Troubleshoot#Petalinux2018.3
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install:$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two. Basic Note to TE Petalinux Templates, see: PetaLinux TE-Template#Template-PetaLinux2018.3

1. PetaLinux Working Environment (see UG1144)
   - PetaLinux Tools Installation is completed.
   - "/bin/sh" is bash
   - Set Working Environment:
     - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
   - $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**
     - <CPU_TYPE>: zynqMP, zynq, microblaze
     - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
   - Export Hardware Definition File (*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
     - change to PetaLinux root folder:
     - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-config**
   - Run:$ **petalinux-config -c u-boot**
   - Run:$ **petalinux-config -c kernel**
   - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with SDK/HSI tools.

Petalinux Configuration

Most settings can be changed with menu-config:

- $ **petalinux-config**
- $ **petalinux-config -c u-boot**

- $ **petalinux-config -c kernel**
- $ **petalinux-config -c rootfs**

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-bsp/device-tree/files/system-user.dtsi recipes-bsp/device-tree/files/zynqmp-qemu-arm.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with "**petalinux-config -c kernel**" will be add here automatically
Apps	project-spec/meta-user/recipes-apps	add simple new app with "**petalinux-create -t apps -n myapp --enable**" enable/disable with "**petalinux-config -c rootfs**"

PetaLinux 2018.2

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2018.2 with SDK 2018.2 and PetaLinux 2018.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Change owner of installation directory to non root (see Troubleshoot page)
  2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  3. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  4. Deactivate Webtalk: $ **petalinux-util --webtalk off**
  5. See also Petalinux Troubleshoot - Petalinux2018.2
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install:$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

Some 18.2 designs include "./**init_config.sh**" to change directory path (CONFIG_TMP_DIR_LOCATION on <plnx-proj-root> /project-spec/configs/config). This is not longer necessary with petalinux 2018.2. this files will be removed on later updates

1. PetaLinux Working Environment (see UG1144)
   - PetaLinux Tools Installation is completed.
   - "/bin/sh" is bash
   - Set Working Environment:
     - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
   - $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**

        - <CPU_TYPE>: zynqMP, zynq, microblaze
        - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
    - Export Hardware Definition File (*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
        - change to PetaLinux root folder:
        - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-config**
    - Run:$ **petalinux-config -c kernel**
    - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with SDK/HSI tools.

Petalinux Configuration

Most settings can be changed with menu-config ($ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-bsp/device-tree/files/system-user.dtsi recipes-bsp/device-tree/files/zynqmp-qemu-arm.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with "**petalinux-config -c kernel**" will be add here automatically

PetaLinux 2017.4

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
    - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
    - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
        - Example: Use Vivado 2017.4 with SDK 2017.4 and PetaLinux 2017.4
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
    1. Change owner of installation directory to non root (see Troubleshoot page)
    2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
    3. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
    4. Deactivate Webtalk: $ **petalinux-util --webtalk off**
    5. See also Petalinux Troubleshoot - Petalinux2017.4
- Note:
    - There is no need to install anything else, or to fetch anything from any github repos, etc.
    - It is recommended to test the installation by creating a dummy template project and building it.
        - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
        - Install:$ **petalinux-create -t project -s <path-to-bsp>**
        - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

PetaLinux 17.4 is using one absolute path in "<plnx-proj-root>/project-spec/configs/config
Change path of CONFIG_TMP_DIR_LOCATION variable to your project path "<plnx-proj-root>/build/tmp" manually or use provided "./**init_config.sh**" to change path variable automatically. If missing, change execution rights for init_config.sh with **chmod**

1. PetaLinux Working Environment (see UG1144)
    - PetaLinux Tools Installation is completed.
    - "/bin/sh" is bash
    - Set Working Environment:
        - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
    - $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**
        - <CPU_TYPE>: zynqMP, zynq, microblaze
        - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
    - Export Hardware Definition File (*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
        - change to PetaLinux root folder:
        - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-config**
    - Run:$ **petalinux-config -c kernel**
    - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with SDK/HSI tools.
7. Note: uboot need also boot.scr from the "<plnx-proj-root>/images/linux" folder. This can be put on SD card or must be included into Boot.bin for QSPI boot without SD usage.

Petalinux Configuration

Most settings can be changed with menu-config ($ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-bsp/device-tree/files/system-user.dtsi recipes-bsp/device-tree/files/zynqmp-qemu-arm.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with "**petalinux-config -c kernel**" will be add here automatically

PetaLinux 2017.2

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
    - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.

- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2017.2 with SDK 2017.2 and PetaLinux 2017.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Change owner of installation directory to non root (see Troubleshoot page)
  2. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  3. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  4. See Troubleshoot Petalinux2017.2
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install:$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

PetaLinux 17.2 is using one absolute path in "<plnx-proj-root>/project-spec/configs/config
Change path of CONFIG_TMP_DIR_LOCATION variable to your project path "<plnx-proj-root>/build/tmp" manually or use provided "**init_config.sh**" to change path variable automatically.

1. PetaLinux Working Environment (see UG1144)
   - PetaLinux Tools Installation is completed.
   - "/bin/sh" is bash
   - Set Working Environment:
     - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
   - $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**
     - <CPU_TYPE>: zynqMP, zynq, microblaze
     - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
   - Export Hardware Definition File (*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
     - change to PetaLinux root folder:
     - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-config**
   - Run:$ **petalinux-config -c kernel**
   - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL and PMU Firmware (ZynqMP only) with SDK/HSI tools.

Petalinux Configuration

Most settings can be changed with menu-config ($ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings

Device Tree	recipes-bsp/device-tree/files/system-user.dtsi recipes-bsp/device-tree/files/zynqmp-qemu-arm.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with "**petalinux-config -c kernel**" will be add here automatically

PetaLinux 2017.1

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2017.1 with SDK 2017.1 and PetaLinux 2017.1
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  2. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  3. See Troubleshoot Petalinux2017.1
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install:$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

PetaLinux 17.1 is using one absolute path in "<plnx-proj-root>/project-spec/configs/config
Change path of CONFIG_TMP_DIR_LOCATION variable to your project path "<plnx-proj-root>/build/tmp" manually.

1. PetaLinux Working Environment (see UG1144)
   - PetaLinux Tools Installation is completed.
   - "/bin/sh" is bash
   - Set Working Environment:
     - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
   - $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**
     - <CPU_TYPE>: zynqMP, zynq, microblaze
     - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
   - Export Hardware Definition File (*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
     - change to PetaLinux root folder:
     - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
   - While anywhere in the project folder tree:

- Run:$ **petalinux-config**
- Run:$ **petalinux-config -c kernel**
- Run:$ **petalinux-config -c rootfs**

5. Build System Image (see UG1144):
   - While anywhere in the project folder tree:
   - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub, bl31.elf (ZynqMP only) and pmufw.elf (ZynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL with SDK/HSI tools.

Petalinux Configuration

Most settings can be changed with menu-config ($ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-bsp/device-tree/files/system-user.dtsi	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with "**petalinux-config -c kernel**" will be add here automatically

PetaLinux 2016.4

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
  - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
  - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
    - Example: Use Vivado 2016.4 with SDK 2016.4 and PetaLinux 2016.4
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
  1. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
  2. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
  3. See Troubleshoot Petalinux2016.4
- Note:
  - There is no need to install anything else, or to fetch anything from any github repos, etc.
  - It is recommended to test the installation by creating a dummy template project and building it.
    - Download one of the BSP Examples from Xilinx Website (Only to test your installation)
    - Install:$ **petalinux-create -t project -s <path-to-bsp>**
    - Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

PetaLinux 16.4 is using one absolute path in "<plnx-proj-root>/project-spec/configs/config
Change path of CONFIG_TMP_DIR_LOCATION variable to your project path "<plnx-proj-root>/build/tmp" manually.

1. PetaLinux Working Environment (see UG1144)
    - PetaLinux Tools Installation is completed.
    - "/bin/sh" is bash
    - Set Working Environment:
        - $ **source <path-to-installed-PetaLinux>/settings.sh**
2. Create a New Project (see UG1144):
    - $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**
        - <CPU_TYPE>: zynqMP, zynq, microblaze
        - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
    - Export Hardware Definition File (*.hdf) from the Vivado Project into the PetaLinux root folder "<plnx-proj-root>":
        - change to PetaLinux root folder:
        - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-config**
    - Run:$ **petalinux-config -c kernel**
    - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-build**
6. Take u-boot.elf, image.ub and bl31.elf (zynqMP only) from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL with SDK/HSI tools.

Petalinux Configuration

Most settings can be changed with menu-config ($ **petalinux-config**, **petalinux-config -c kernel**, **petalinux-config -c rootfs**).

Manual changes can be done in the subfolder "<plnx-proj-root>/project-spec/meta-user/"

U-Boot	recipes-bsp/u-boot/files/platform-top.h	overwrite, add UBoot settings
Device Tree	recipes-dt/device-tree/files/system-top.dts	overwrite, add device tree attributes
Kernel	recipes-kernel/linux/linux-xlnx/	changes with "**petalinux-config -c kernel**" will be add here automatically

PetaLinux 2016.2

PetaLinux Installation

- (optional) Create new VM with supported Linux OS.
    - **Attention:** Use English as OS language for your Linux System (Keyboard language can be any language). Other languages may cause errors on PetaLinux build process.
- Download PetaLinux from Xilinx Website: http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html
    - Choose a PetaLinux Version, that's corresponding to the installed Vivado and SDK Version.
        - Example: Use Vivado 2016.2 with SDK 2016.2 and PetaLinux 2016.2
- Use UG1144 "PetaLinux Tools Documentation - Reference Guide" that's corresponding with your PetaLinux Version
    1. Check "PetaLinux Tools Installation Requirements" chapter and install missing tool/libraries
    2. Use installation instructions from chapter "PetaLinux Tools Installation Steps"
    3. Additional packages for PetaLinux 2016.2 + MicroBlaze projects:
        - $ **sudo apt-get install libc6-i386 lib32stdc++6 lib32gcc1 lib32ncurses5 lib32z1**
- Note:
    - There is no need to install anything else, or to fetch anything from any github repos, etc.
    - It is recommended to test the installation by creating a dummy template project and building it.
        - Download one of the BSP Examples from Xilinx Website (Only to test your installation)

- Install:$ **petalinux-create -t project -s <path-to-bsp>**
- Build: $ **petalinux-build**

Creating a Project from Vivado Project

Some reference designs contains a preconfigured PetaLinux project as template. This can be used instead of creating a new project described on step two.

1. PetaLinux Working Environment (see UG1144)
    - PetaLinux Tools Installation is completed.
    - "/bin/sh" is bash
    - Set Working Environment:
        - $ **source <path-to-installed-PetaLinux>/settings.sh**
    - Set cross compiler:
        - $ **export CROSS_COMPILE=arm-xilinx-linux-gnueabi-**
        - $ **export ARCH=arm**
2. Create a New Project (see UG1144):
    - $ **petalinux-create --type project --template <CPU_TYPE> --name <PROJECT_NAME>**
        - <CPU_TYPE>: zynqMP, zynq, microblaze
        - <PROJECT_NAME>:The name of the project you are building
3. Import Hardware Configuration (see UG1144):
    - Export Hardware Definition File (*.hdf) from the Vivado Project into the PetaLinux subfolder "<plnx-proj-root>/hw-description":
        - change to the hardware description folder:
        - Run:$ **petalinux-config --get-hw-description**
4. (optional) Configure your PetaLinux:
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-config**
        - It's recommended to deactivate FSBL-Configuration and build FSBL and Boot.bin with SDK/HSI: "linux Components Selection --->" "[ ] First Stage Bootloader
    - Run:$ **petalinux-config -c kernel**
    - Run:$ **petalinux-config -c rootfs**
5. Build System Image (see UG1144):
    - While anywhere in the project folder tree:
    - Run:$ **petalinux-build**
    - Build log-file:"<plnx-proj-root>/build/build.log"
6. Take u-boot.elf and image.ub from "<plnx-proj-root>/images/linux" and make BOOT.BIN. It is recommended to create the FSBL with SDK /HSI tools.

Petalinux Configuration

Most settings can be changed with menu-config ($ **petalinux-config**).

There are 3 more files that user can edit, they are

File	Description	Location
system-top.dts	Device tree changes are to be applied here	<petalinux-project>/subsystems/linux/configs/device-tree
platform-top.h	U-boot changes are to be applied here	<petalinux-project>/subsystems/linux/configs/u-boot
fsbl_hooks.c	FSBL changes are to be applied here	SDK/HSI FSBL-Template

There should be no reason to modify any other files by editing them (most of them are generated and would be overwritten).