PetaLinux TE-Template

- PetaLinux 2023.2
  - Create own template
  - Use template
  - Issues and Notes
- PetaLinux 2022.2
  - Create own template
  - Use template
  - Issues and Notes
- PetaLinux 2021.2
  - Create own template
  - Use template
  - Issues and Notes
- PetaLinux 2020.2
  - Create own template
  - Use template
  - Issues and Notes
- PetaLinux 2019.2
  - Create own template
  - Use template
  - Issues and Notes
- PetaLinux 2018.3
  - Create own template
  - Use template
  - Issues and Notes
- PetaLinux 2018.2
  - Create own template
  - Use template
  - Issues and Notes
- PetaLinux 2017.4
  - Create own template
  - Use template
  - Issues and Notes

PetaLinux 2023.2



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

Create own template

1. Clear existing project with: **petalinux-build -x mrproper**
   a. **in** case changes in u-boot or kernel, see Issues and note
2. change/remove manually(on linux file manager press Strg+H to see hidden files):

folder/file	Delete	Note
<base folder>	• *.xsa • .gitignore	"config.project" must be stay there
.petalinux/	all, excepted "metadata"	"metadata" must be stay there
.xil/	complete folder	
components/	---	should be empty

project-spec/	• yocto-layer.log	"attributes" must be stay there
project-spec/configs/	• *.old • .Xil	"config" and "rootfs_config" must be stay there
project-spec/hw-description/	all, excepted "metadata"	"metadata" must be stay there
project-spec/meta-user	---	complete folder must be stay there

3. uncomment DDR settings in ""project-spec/configs/config", see Issues and notes

Use template

1. copy xsa into base folder
2. reload XSA with **petalinux-config --get-hw-description**
3. build or change design like prefer

Issues and Notes

Issue	Workaround	Note
DDR size will be not reloaded	comment out DDR settings in "project-spec/configs/config" to force DDR reload with XSA import	Only important, if DDR size will is changed

PetaLinux 2022.2

Create own template

1. Clear existing project with: **petalinux-build -x mrproper**
    a. **in** case changes in u-boot or kernel, see Issues and note
2. change/remove manually(on linux file manager press Strg+H to see hidden files):

folder/file	Delete	Note
<base folder>	• *.xsa • .gitignore	"config.project" must be stay there
.petalinux/	all, excepted "metadata"	"metadata" must be stay there
.xil/	complete folder	
components/	---	should be empty
project-spec/	• yocto-layer.log	"attributes" must be stay there
project-spec/configs/	• config.old • rootfs_config.old	"config" and "rootfs_config" must be stay there
project-spec/hw-description/	all, excepted "metadata"	"metadata" must be stay there

project-spec/meta-user	---	complete folder must be stay there

3. uncomment DDR settings in ""project-spec/configs/config", see Issues and notes

Use template

1. copy xsa into base folder
2. reload XSA with **petalinux-config --get-hw-description**
3. build or change design like prefer

Issues and Notes

Issue	Workaround	Note
DDR size will be not reloaded	comment out DDR settings in "project-spec/configs/config" to force DDR reload with XSA import	Only important, if DDR size will is changed

PetaLinux 2021.2

Petalinux 2021.2 is under review documentation for 2021.2 can be changed permanently at the moment

Create own template

1. Clear existing project with: **petalinux-build -x mrproper**
    a. **in** case changes in u-boot or kernel, see Issues and note
2. change/remove manually(on linux file manager press Strg+H to see hidden files):

folder/file	Delete	Note
<base folder>	• *.xsa • .gitignore	"config.project" must be stay there
.petalinux/	all, excepted "metadata"	"metadata" must be stay there
.xil/	complete folder	
components/	---	should be empty
project-spec/	• yocto-layer.log	"attributes" must be stay there
project-spec/configs/	• config.old • rootfs_config.old	"config" and "rootfs_config" must be stay there
project-spec/hw-description/	all, excepted "metadata"	"metadata" must be stay there
project-spec/meta-user	---	complete folder must be stay there

3. uncomment DDR settings in ""project-spec/configs/config", see Issues and notes

Use template

1. copy xsa into base folder
2. reload XSA with **petalinux-config --get-hw-description**
3. build or change design like prefer

Issues and Notes

Issue	Workaround	Note
DDR size will be not reloaded	comment out DDR settings in "project-spec/configs/config" to force DDR reload with XSA import	Only important, if DDR size will is changed

PetaLinux 2020.2

Create own template

1. Clear existing project with: **petalinux-build -x mrproper**
    a. **in** case changes in u-boot or kernel, see Issues and note
2. change/remove manually(on linux file manager press Strg+H to see hidden files):

folder/file	Delete	Note
<base folder>	• *.xsa • .gitignore	"config.project" must be stay there
.petalinux/	all, excepted "metadata"	"metadata" must be stay there
.xil/	complete folder	
components/	---	should be empty
project-spec/	• yocto-layer.log	"attributes" must be stay there
project-spec/configs/	• config.old • rootfs_config.old	"config" and "rootfs_config" must be stay there
project-spec/hw-description/	all, excepted "metadata"	"metadata" must be stay there
project-spec/meta-user	---	complete folder must be stay there

3. uncomment DDR settings in ""project-spec/configs/config", see Issues and notes

Use template

1. copy xsa into base folder
2. reload XSA with **petalinux-config --get-hw-description**
3. build or change design like prefer

Issues and Notes

Issue	Workaround	Note
DDR size will be not reloaded	comment out DDR settings in "project-spec/configs/config" to force DDR reload with XSA import	Only important, if DDR size will is changed

PetaLinux 2019.2

Create own template

1. Clear existing project with: **petalinux-build -x mrproper**
    a. **in** case changes in u-boot or kernel, see Issues and note
2. change/remove manually:

folder/file	Delete	Note
<base folder>	• *.xsa • .gitignore	"config.project" must be stay there
.petalinux/	all, excepted "metadata"	"metadata" must be stay there
.xil/	complete folder	
components/	---	should be empty
project-spec/	• yocto-layer.log	"attributes" must be stay there
project-spec/configs/	• config.old • rootfs_config.old	"config" and "rootfs_config" must be stay there
project-spec/hw-description/	all, excepted "metadata"	"metadata" must be stay there
project-spec/meta-plnx-generated	complete folder	
project-spec/meta-user	---	complete folder must be stay there

Use template

1. copy hdf into base folder
2. reload XSA with **petalinux-config --get-hw-description**
3. build or change design like prefer

Issues and Notes

Issue	Workaround	Note

DDR size will be not reloaded	comment out DDR settings in "project-spec/configs /config" to force DDR reload with HDF import	Only important, if DDR size will is changed
Kernel and u–bot changes will not saved automatically in meta-user	`run:` • `petalinux-build -c kernel -x finish -f` • `petalinux-build -c u-boot -x finish -f` before the project will be cleared.	`only need if you change something in uboot ot kernel setup`

PetaLinux 2018.3

Create own template

1. Clear existing project with: **petalinux-build -x mrproper**
2. change/remove manually:

folder/file	Delete	Note
<base folder>	• *.hdf • .gitignore	"config.project" must be stay there
.petalinux/	all, excepted "metadata"	"metadata" must be stay there
.xil/	complete folder	
components/	---	should be empty
project-spec/	• yocto-layer.log	"attributes" must be stay there
project-spec/configs/	• config.old • rootfs_config.old	"config" and "rootfs_config" must be stay there
project-spec/hw-description/	all, excepted "metadata"	"metadata" must be stay there
project-spec/meta-plnx-generated	complete folder	
project-spec/meta-user	---	complete folder must be stay there

Use template

1. copy hdf into base folder
2. reload HDF with **petalinux-config --get-hw-description**
3. build or change design like prefer

Issues and Notes

Issue	Workaround	Note

DDR size will be not reloaded	comment out DDR settings in "project-spec/configs/config" to force DDR reload with HDF import	Only important, if DDR size will is changed

PetaLinux 2018.2

Create own template

1. Clear existing project with: **petalinux-build -x mrproper**
2. change/remove manually:

folder/file	Delete	Note
<base folder>	• *.hdf • .gitignore	"config.project" must be stay there
.petalinux/	all, excepted "metadata"	"metadata" must be stay there
.xil/	complete folder	
components/	---	should be empty
project-spec/	• yocto-layer.log	"attributes" must be stay there
project-spec/configs/	• config.old • rootfs_config.old	"config" and "rootfs_config" must be stay there
project-spec/hw-description/	all, excepted "metadata"	"metadata" must be stay there
project-spec/meta-plnx-generated	complete folder	
project-spec/meta-user	---	complete folder must be stay there

Use template

1. copy hdf into base folder
2. reload HDF with **petalinux-config --get-hw-description**
3. build or change design like prefer

Issues and Notes

Issue	Workaround	Note
DDR size will be not reloaded	comment out DDR settings in "project-spec/configs/config" to force DDR reload with HDF import	Only important, if DDR size will is changed

PetaLinux 2017.4

Create own template

1. Clear existing project with: **petalinux-build -x mrproper**
2. change/remove manually:

folder/file	Delete Files/Folder	Note
<base folder>	• *.hdf • .gitignore	"config.project" must be stay there
.petalinux/	all, excepted "metadata"	"metadata" must be stay there
.xil/	complete folder	
components/	---	should be empty
project-spec/	• yocto-layer.log	"attributes" must be stay there
project-spec/configs/	• config.old • rootfs_config.old	"config" and "rootfs_config" must be stay there
project-spec/hw-description/	all, excepted "metadata"	"metadata" must be stay there
project-spec/meta-plnx-generated	complete folder	
project-spec/meta-user	---	complete folder must be stay there

Use template

1. copy hdf into base folder
2. reload HDF with **petalinux-config --get-hw-description**
3. build or change design like prefer

Issues and Notes

Issue	Workaround	Note
temp folder path is absolute path	change CONFIG_TMP_DIR_LOCATION from "project-spec /configs/config" to the correct project dir	Trenz Electronic provide init_config.sh script to change path to the current project