Release Description

Last supported Release

Type or File	Version
Vivado Design Suite	2015.4
Trenz Project Scripts	2015.4.26
Trenz <board_series>_board_files.csv	1.2
Trenz apps_list.csv	1.6
Trenz zip_ignore_list.csv	1.0

Name Description of Zip-delivery

Description	PCB Name		Project Name+(opt. Variant)		supported VIVADO Version		Build Version and Date
Example:	TE0726	-	test_board_noprebuilt	-	vivado_2015.4	-	build_26_20160415133543

Directory structure

File or Directory	Type	Description
<design_name>	base directory	Base directory with predefined batch files (*.cmd) to generate or open VIVADO-Project
<design_name> /block_design/	source	Script to generate Block Design in Vivado (*_bd.tcl). (optional) Some board part designs used subfolder <board_file_shortname> with Board Part specific Block Design (*_bd.tcl).
<design_name> /board_files/	source	Local board part files repository and a list of available board part files (<board_series>_board_files.csv)
<design_name> /constraints/	source	Project constrains (*.xdc). Some board part designs used subfolder <board_file_shortname> with additional constrains (*.xdc)
<design_name> /doc/	source	Documentation
<design_name> /ip_lib/	source	Local Vivado IP repository
<design_name> /misc/	source	(Optional) Directory with additional sources
<design_name> /prebuilt /boot_images/	prebuilt	Directory with prebuilt boot images (*.bin) and configuration files (*.bif) for zynq and configured hardware files (*.bit and *.mcs) for micoblaze included in subfolders: default or <board_file_shortname> /<app_name>
<design_name> /prebuilt /hardware/	prebuilt	Directory with prebuilt hardware sources (*.bit, *hdf, *.mcs) and reports included in subfolders: default or <board_file_shortname>
<design_name> /prebuilt /software/	prebuilt	(Optional) Directory with prebuilt software sources (*.elf) included in subfolders: default or <board_file_shortname>/<app_name>
<design_name> /prebuilt/os/	prebuilt	(Optional) Directory with predefined OS images included in subfolders <os_name>/<board_file_shortname> or <os_name> /default
<design_name> /scripts/	source	TCL scripts to build a project

File or Directory	Type	Description
<design_name> /settings/	source	(Optional) Additional design settings: zip_ignore_list.csv, vivado project settings, SDSOC settings
<design_name> /software/	source	(Optional) Directory with additional software
<design_name> /sw_lib/	source	(Optional) Directory with local SDK/HSI software IP repository and a list of available software (apps_list.csv)
<design_name> /v_log/	generated	Temporary directory with vivado log files (used only when vivado is started with predefined command files (*.cmd) from base folder otherwise this logs will be writen into the vivado working directory)
<design_name> /vivado/	work, generated	Working directory where Vivado project is created. Vivado project file is <design_name>.xpr
<design_name> /vivado_lab/	work, generated	(Optional) Working directory where Vivado LabTools is created. LabTools project file is <design_name>.lpr
<design_name> /workspace/hsi	work, generated	(Optional) Directory where hsi project is created
<design_name> /workspace/sdk	work, generated	(Optional) Directory where sdk project is created
<design_name> /sdsoc	work, generated	(Optional) Directory where SDSOC project is created
<design_name> /backup/	generated	(Optional) Directory for project backups

Windows Command Files

File Name	Description
Design + Settings	
design_basic_settings.cmd	Settings for the other *.cmd files. Following Settings are avaliable:

File Name	Description

	• General Settings:
	  • (optional) **DO_NOT_CLOSE_SHELL**: Shell do not closed after processing
	  • (optional) **ZIP_PATH**: Set Path to installed Zip-Program. Currently 7-Zip are supported. IUsed fc predefined TCL-function to Backup project.
	  • (optional) **ENABLE_SDSOC**: Enable SDSOC Setting. Currently only for some reference projec as beta version!
	• Xilinx Setting:
	  • **XILDIR**: Set Xilinx installation path (Default: c:\Xilinx).
	  • **VIVADO_VERSION**: Current Vivado/LabTool /SDK Version (Example:2015.4). Don't change Vivado Version.
	  • Xilinx Software will be searched in:
	  • VIVADO (optional for project creation and programming): %XILDIR%\Vivado\% VIVADO_VERSION%\
	  • SDK (optional for software projects and programming): %XILDIR%\SDK\% VIVADO_VERSION%\
	  • LabTools (optional for programming only): %XILDIR%\Vivado_Lab\% VIVADO_VERSION%\
	  • SDSOC (optional): %XILDIR%\SDSOC\% VIVADO_VERSION%\

File Name	Description

	• Board Setting:
	  • **PARTNUMBER**: Set Board part number of the project which should be created
	  • Available Numbers: (you can use ID, PRODID,BOARDNAME or SHORTNAME from TExxxx_board_file.csv list)
	  • Used for project creation and programmin
	  • To create empty project without board par used PARTNUMBER=-1 (use GUI to create your project. No block design tcl-file should be in /block_design)
	  • Example TE0726 Module :
	  • USE ID |USE PRODID |Use Boardname |Use Shortname PARTNUMBER=1|PARTNUMBER=te0726 01 |PARTNUMBER=trenz.biz:te0726-01: part0:1.0 |PARTNUMBER=TE0726-01
	• Programming Settings(program*file.cmd):
	  • **SWAPP**: Select Software App, which should be configured.
	  • Use the folder name of the <design_name /prebuilt/boot_image/<partname>/* subfolder. The *bin,*.mcs or *.bit from this folder will be used.
	  • If you will configure the raw *.bit or *.mcs bin from the <design_name>/prebuilt /hardware/<partname>/ folder, use @set SWAPP=NA or @set SWAPP="".
	  • Example: SWAPP=hello_world used the file from prebuilt/boot_image/<partname> /hello_world
	SWAPP=NA used the file from <design_name>/prebuilt/boot_image /<partname>/
	  • **PROGRAM_ROOT_FOLDER_FILE**: If you want to program design file from the rootfolder <design_name>, set to 1
	  • Attention: it should be only one *.bit, *.msc or *.bin file in the root folder.

File Name	Description
design_clear_design_folders.cmd	(optional) Attention: Delete "<design_name>/v_log/", "<design_name>/vivado/", "<design_name>/vivado_lab/", "<design_name>/sdsoc/", and "<design_name>/workspace/" directory with related documents!
design_run_project_batchmode.cmd	(optional) Create Project with setting from "design_basic_settings.cmd" and source folders. Build all Vivado hardware and software files if the sources are available Delete "<design_name>/vivado/", and "<design_name>/workspace/hsi/" directory with related documents before Projekt will created.
Hardware Design	
vivado_create_project_guimode.cmd	Create Project with setting from "design_basic_settings.cmd" and source folders. Vivado GUI will be opened during the process. Delete "<design_name>/vivado/", and "<design_name>/workspace/" directory with related documents before Projekt will created.
vivado_create_project_batchmode.cmd	(optional) Create Project with setting from "design_basic_settings.cmd" and source folders. Delete "<design_name>/vivado/", and "<design_name>/workspace/" directory with related documents before Projekt will created.
vivado_open_existing_project_guimode.cmd	Opens an existing Project "<design_name>/vivado/<design_name>.xpr" and restore Script-Variables.
Software Design	
sdk_create_prebuilt_project_guimode.cmd	(optional) Create SDK project with hardware definition file from prebuild folder. It used the *.hdf from: <design_name>/prebuilt/hardware/<board_file_shortname>/. Set <board_file_shortname> and <app_name> in "design_basic_settings.cmd".
Programming	

File Name	Description
program_flash_binfile.cmd	(optional) For Zynq Systems only. Programming Flash Memory via JTAG with specified Boot.bin. Used SDK Programmer (Same as SDK "Program Flash") or LabTools Programmer (Vivado or LabTools only), depends on installion settings. Default, it used the boot.bin from: <design_name>/prebuilt /boot_images/<board_file_shortname>/<app_name>. Settings are done in "design_basic_settings.cmd".
program_flash_mcsfile.cmd	(optional) For Non-Zynq Systems only. Programming Flash Memory via JTAG with specified <design_name>.mcs. Used LabTools Programmer (Vivado or LabTools only), depends on installion settings. Default, it used the <design_name>.mcs from: <design_name>/prebuilt/hardware /<board_file_shortname>. Settings are done in "design_basic_settings.cmd".
program_fpga_bitfile.cmd	(optional) Programming FPGA via JTAG with specified <design_name>.bit. Used LabTools Programmer (Vivado or LabTools only), depends on installion settings. Default, it used the <design_name>.bit from: <design_name>/prebuilt/hardwa /<board_file_shortname>. Settings are done in "design_basic_settings.cmd".
labtools_open_project_guimode.cmd	(optional) Create or open an existing Vivado Lab Tools Project (Additional TCL functions from Programming and Utilities Group are usable). Settings are done in "design_basic_setting cmd".

Additional TCL functions for usage on Vivado TCL Console

Name	Options	Description (Default Configuration)
TE::help		Display currently available functions. Important: Use only displayed functions and no functions from sub-namespaces
Hardware Design		

Name	Options	Description (Default Configuration)
TE::hw_blockdesign_export_tcl	[-no_mig_contents] [-no_validate] [-mod_tcl] [-svntxt <arg>] [-board_part_only] [-help]	Export Blockdesign to project folder <design_name>/block_design/ . Old *bd.tcl will be overwritten!
TE::hw_build_design	[-export_prebuilt] [-export_prebuilt_only] [-help]	Run Synthese, Implement, and generate Bitfile, optional MCSfile and some report files
Software Design		
TE::sw_run_hsi	[-run_only] [-prebuilt_hdf <arg>] [-no_hsi] [-no_bif] [-no_bin] [-no_bitmcs] [-clear] [-help]	Copies current Hardware files and reports from the vivado project to the prebuilt folder, if -prebuild_hdf <arg> isn't set. Copy the Hardware Defintition file to the working directory:<design_name>/workspace/hsi Run HSI in <design_name>/workspace/hsi for all Programes listed in <design_name>/sw_lib/apps_list.csv If HSI is finished, BIF-GEN and BIN-Gen are running for these Apps in the prepuilt folders <design_name>/prebuilt/... You can deactivate different steps with following args : • -no_hsi : *.elf filesgeneration is disabled • -no_bif : *.bif files generation is disabled • -no_bin : *.bin files generation is disabled • -no_bitmcs: *.bit and *.mcs file (with software design) is disabled

Name	Options	Description (Default Configuration)
TE::sw_run_sdk	[-open_only] [-update_hdf_only] [-prebuilt_hdf <arg>] [-clear] [-help]	Copies current Hardware files and reports from the vivado project to the prebuilt folder, if -prebuild_hdf <arg> isn't set. Copy the Hardware Defintition file to the working directory:<design_name> /workspace/sdk Start SDK GUI in this workspace
Programming		
TE:: pr_init_hardware_manager	[-help]	Open Hardwaremanager, autoconnect target device and initialise flash memory with configuration from *_board_files.csv.
TE:: pr_program_jtag_bitfile	[-used_board <arg>] [-swapp <arg>] [-available_apps] [-used_basefolder_bitfile] [-help]	Copies current Hardware files and reports from the vivado project to the prebuilt folder, if -used_board <arg> isn't set (Vivado only). Programming Bitfile from <design_name> /prebuilt/hardware /<board_file_shortname> to the fpga device. If "-used_basefolder_bitfile" is set, the Bitfile (*.bit) from the base folder (<design_name>) is used instead of the prebuilts. Attention: Take only one Bitfile in the basefolder! (MicroBlaze only) If "-swapp" is set, the Bitfile with *.elf configuration is used from <design_name>/prebuilt/boot_images /<board_file_shortname>/<app_name>

Name	Options	Description (Default Configuration)
TE::pr_program_flash_binfile	[-no_reboot] [-used_board <arg>] [-swapp <arg>] [-available_apps] [-force_hw_manager] [-used_basefolder_binfile] [-help]	Attention: For Zynq Systems only! Program the Bootbin from <design_name>/prebuilt/boot_images/<board_file_shortname>/<app_name> to the fpga device. Appname is selected with: -swapp <app_name> After programming device reboot from memory will be done. Default SDK Programmer is used, if not available LabTools Programmer is used. If "-used_basefolder_binfile" is set, the Binfile (*.bin) from the base folder (<design_name>) is used instead of the prebuilts. Attention: Take only one Binfile in the basefolder!
TE::pr_program_flash_mcsfile	[-no_reboot] [-used_board <arg>] [-swapp <arg>] [-available_apps] [-used_basefolder_mcsfile] [-help]	Copies current Hardware files and reports from the vivado project to the prebuilt folder, if -used_board <arg> isn't set (Vivado only). Initialise flash memory with configuration from *_board_files.csv Programming MCSfile from <design_name>/prebuilt/hardware/<board_file_shortname> to the Flash Device. After programming device reboot from memory will be done. If "-used_basefolder_binfile" is set, the MCSfile (*.mcs) from the base folder (<design_name>) is used instead of the prebuilts. Attention: Take only one MCSfile in the basefolder! (MicroBlaze only) If "-swapp" is set, the MCSfile with *.elf configuration is used from <design_name>/prebuilt/boot_images/<board_file_shortname>/<app_name>
Utilities		

Name	Options	Description (Default Configuration)
TE::util_zip_project	[-save_all] [-remove_prebuilt] [-manual_filename <arg>] [-help]	Make a Backup from your Project in <design_name>/backup/ Zip-Program Variable must be set in start_settings.cmd. Currently only 7-Zip is supported.
Beta Test		
TE::beta_util_sdsoc_project	[-check_only] [-start_sdsoc_only] [-help]	Create SDSO Workspace. Currently only on some Reference Designs available. Run [-check_only] to check SDSOC ready state.

Usage

Create project

1. Unzip project files
2. Rename basefolder (basefolder name is used as project name)
3. Edit design_basic_settings.cmd
    a. Select the correct Xilinx Program path (See: **Windows Command Files** design_basic_settings .cmd)
    b. Select the correct board part number for your PCB (See: **Windows Command Files** design_basic_settings .cmd)
    c. Other settings are optional (See: **Windows Command Files** design_basic_settings .cmd)
4. Excecute vivado_create_project_guimode.cmd or vivado_create_project_batchmode.cmd to generate a vivado project with the predefined Block Design from the Block Design folder
5. Open Vivado with vivado_open_existing_project_guimode.cmd (if you use vivado_create_project_guimode .cmd on step 4, you didn't need this)
6. Open the Block Design and create your own design inside this Block Design.
7. Backup your Block Design as tcl-script: Type "TE::hw_blockdesign_export_tcl" on Vivado Tcl Console. The old one will be overwritten.
8. Build your Design...

Initial scripts on existing Vivado/LabTool project

- Variant 1 (recommended):
  - Start the project with the predefined command file (vivado_open_existing_project_guimode.cmd .cmd) respectively LabTools with (labtools_open_project_guimode.cmd)
- Variant 2:
  - Create your own Initialisation Button on the Vivado GUI:
    - Tools Customize Commands Customize Commands...
    - Push 
    - Type Name ex.: Init Scripts
    - Press Enter
    - Select Run command and insert :
      - for Vivado: cd [get_property DIRECTORY [current_project]]; source -notrace "../scripts/reinitialise_all.tcl"
      - for LabTool: cd [pwd]; source -notrace "../scripts/reinitialise_all.tcl"
    - Press Enter
    - A new Button is shown on the Vivado Gui: All Scripts are reinitialised, if you press this Button.
- Variant 3:
  - Reinitialise Script on Vivado TCL-Console:
    - Type: source ../scripts/reinitialise_all.tcl

Use predefined TE-Script functions

- Variant 1 (recommended):
  - Typ function on Vivado TCL Console, ex.: TE::help
  - TE::help
    - Show all predefined TE-Script functions.
  - TE:<functionname> -help
    - Show short description of this function.
    - Attention: If -help argument is set, all other args will be ignored.

- Variant 2:
  - Create your own function Button on the Vivado GUI:
    - Tools Customize Commands Customize Commands...
    - Push +
    - Type Name ex.: Run SDK
    - Press Enter
    - Select Run command and insert function:
      - Variante 1 (no Vivado request window for args):
        - insert function and used args, ex.: TE::sw_program_zynq -swapp hello_world
      - Variant 2 (Vivado request window for args):
        - insert function, ex.: TE::sw_program_zynq
        - Press Define Args...
        - For every arg:
          - Push
          - Typ Name, Comment, Default Value and set optional
          - Press Enter
          - Example for args:
          -
            - Push
            - Index, Key Name, -swapp,
            - Push
            - Appname, Arg, hello_world,
    - Press Enter
    - A new Button is shown on the Vivado Gui .

Checklist / Troubleshoot

1. Are you using exactly the same Vivado version? If not then the scripts will not work, no need to try.
2. Ary you using Vivado in Windows PC? Vivado works in Linux also, but the scripts are tested on Windows only.
3. Is you PC OS Installation English? Vivado may work on national versions also, but there have been known problems.
4. Are space character on the project path? Somtimes TCL-Scripts can't handle this correctly. Remove spaces from project path.
5. Did you have the newest reference design build version? Maybe it's only a bug from a older version.
6. Check <design_name>/v_log/vivado.log? If no logfile exist, wrong xilinx paths are set in design_basic_settings.cmd
7. If nothing helps, send a mail to trenz support (support(at)trenz-electronic.de) with subject line "[TE-Reference Designs] ", the complete zip-name from your reference design and the last log file (<design_name>/v_log/vivado.log)

References

1. Vivado Design Suite User Guide - Getting Started (UG910)
2. Vivado Design Suite User Guide - Using the Vivado IDE (UG893)
3. Zynq-7000 All Programmable SoC Software Developers Guide (UG821)
4. SDSoC Environment User Guide - Getting Started (UG1028)
5. SDSoC Environment - User Guide (UG1027)
6. SDSoC Environment User Guide - Platforms and Libraries (UG1146)