TE0726 Test Board

Table of contents

Overview

Zynq PS Design with Linux Example.
Refer to http://trenz.org/te0726-info for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2020.2
- PetaLinux
- SD
- ETH
- USB
- I2C
- Special FSBL for QSPI programming

Revision History

Date	Vivado	Project Built	Authors	Description
2021-11-04	2020.2	TE0726-test_board_noprebuilt-vivado_2020.2-build_8_20211104085813.zip TE0726-test_board-vivado_2020.2-build_8_20211104085759.zip	Mohsen Chamanbaz	• 0001-QSPI-s25fl127_8-2020_2.patch for restart • Added Ethernet interface
2021-08-30	2020.2	TE0726-test_board_noprebuilt-vivado_2020.2-build_7_20210830095228.zip TE0726-test_board-vivado_2020.2-build_7_20210830095218.zip	Mohsen Chamanbaz	• 2020.2 release • Without Etherner interface
2020-04-08	2019.2	TE0726-test_board_noprebuilt-vivado_2019.2-build_10_20200408185842.zip TE0726-test_board-vivado_2019.2-build_10_20200408185804.zip	Mohsen Chamanbaz /John Hartfiel	• changes FSBL flash

2020-03-25	2019.2	TE0726-test_board_noprebuilt-vivado_2019.2-build_8_20200325080535.zip TE0726-test_board-vivado_2019.2-build_8_20200325080528.zip	Mohsen Chamanbaz /John Hartfiel	• script update
2020-02-14	2019.2	TE0726-test_board_noprebuilt-vivado_2019.2-build_5_20200214091531.zip TE0726-test_board-vivado_2019.2-build_5_20200214091442.zip	Mohsen Chamanbaz	• Update to 19.2 • Vitis support • prebuilt binary export on selection guide
2019-12-12	2018.3	te0726-test_board_noprebuilt-vivado_2018.3-build_10_20191211160322.zip te0726-test_board-vivado_2018.3-build_10_20191211160314.zip	Mohsen Chamanbaz	• FSBL update to18.3 • additional linux apps
2018-07-13	2018.2	te0726-test_board_noprebuilt-vivado_2018.2-build_02_20180713155548.zip te0726-test_board-vivado_2018.2-build_02_20180713155535.zip	John Hartfiel	• Changed SDK Notes on FSBL template fro Flash programming
2018-07-11	2018.2	te0726-test_board_noprebuilt-vivado_2018.2-build_02_20180711113737.zip te0726-test_board-vivado_2018.2-build_02_20180711113722.zip	John Hartfiel	• change note for REV01 • no design changes
2018-02-17	2017.4	te0726-test_board-vivado_2017.4-build_08_20180517084735.zip te0726-test_board_noprebuilt-vivado_2017.4-build_08_20180517084604.zip	John Hartfiel	• correction netboot offset for 128MB variant
2018-02-16	2017.4	te0726-test_board-vivado_2017.4-build_06_20180216205357.zip te0726-test_board_noprebuilt-vivado_2017.4-build_06_20180216205410.zip	John Hartfiel	• correction PS REFCLK for 01 variant

2018-01-31	2017.4	te0726-test_board-vivado_2017.4-build_05_20180131115412.zip te0726-test_board_noprebuilt-vivado_2017.4-build_05_20180131115451.zip	John Hartfiel	• initial release 2017.4

Design Revision History

Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version

Flash Programming failed with 19.2	Depending on Flash content Flash programming failed with provided fsbl_flash (Xilinx AR# 70548)2019.2 version	• Option1:
		  ◦ In case Flash is empty, use fsbl_flash on programming GUI
		  ◦ In case Flash is programmed use normal fsbl on programming GUI
		• Option2: use in both case fsbl_flash on programming GUI and Vivado LabTools 2018.3
			--- 	

FSBL/ Kernel Vivado 2020.2	Petalinux does not restart after first booting	use 0001-QSPI-s25fl127_8-2020_2.patch from test_board\os\petalinux\project-spec\meta-user\recipes-kernel\linux\linux-xlnx\	---

Known Issues

Requirements

Software

Software	Version	Note
Vitis	2020.2	needed, Vivado is included into Vitis installation
PetaLinux	2020.2	needed

Software

Hardware

Basic description of TE Board Part Files is available on TE Board Part Files.

Complete List is available on <design name>/board_files/*_board_files.csv

Design supports following modules:

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
te0726-01	01_64MB	REV01	64MB	16MB	NA	NA	NA
te0726-03r	r_128MB	REV03, REV02	128MB	16MB	NA	NA	LPDDR3, without ETH, USB,Camera, HDMI
te0726-03m	m_512MB	REV03, REV02	512MB	16MB	NA	NA	LPDDR3
te0726-03-07s-1c	7s_512MB	REV03, REV02	512MB	16MB	NA	NA	LPDDR3
TE0726-03RJ	r_128MB	REV03, REV02	128MB	16MB	NA	NA	LPDDR3, without ETH, USB,Camera, HDMI
TE0726-03-41C74-Q	r_128MB	REV03, REV02	128MB	16MB	NA	NA	LPDDR3, Customised
TE0726-03-41C74-R	r_128MB	REV03, REV02	128MB	16MB	NA	NA	LPDDR3, without ETH, USB,Camera, HDMI
TE0726-03IM	mi_512MB	REV03, REV02	512MB	16MB	NA	NA	LPDDR3
TE0726-03-11C64-A	7s_512MB	REV03, REV02	512MB	16MB	NA	NA	LPDDR3
TE0726-03-41I64-A	mi_512MB	REV03, REV02	512MB	16MB	NA	NA	LPDDR3
TE0726-03-41C64-A	m_512MB	REV03, REV02	512MB	16MB	NA	NA	LPDDR3

Hardware Modules

Design supports following carriers:

Carrier Model	Notes

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
USB Cable	Connect to USB2 or better USB3 Hub for proper power over USB

Additional Hardware

Content

For general structure and of the reference design, see Project Delivery - AMD devices

Design Sources

Type	Location	Notes

Vivado	<design name>/block_design <design name>/constraints <design name>/ip_lib	Vivado Project will be generated by TE Scripts
Vitis	<design name>/sw_lib	Additional Software Template for Vitis and apps_list.csv with settings automatically for Vitis app generation
PetaLinux	<design name>/os/petalinux	PetaLinux template with current configuration

Design sources

Additional Sources

Type	Location	Notes
init.sh	<design name>/misc/sd/	Additional Initialization Script for Linux

Additional design sources

Prebuilt

File	File-Extension	Description
BIF-File	*.bif	File with description to generate Bin-File
BIN-File	*.bin	Flash Configuration File with Boot-Image (Zynq-FPGAs)
BIT-File	*.bit	FPGA (PL Part) Configuration File
Diverse Reports	---	Report files in different formats
Hardware-Platform-Specification-Files	*.xsa	Exported Vivado Hardware Specification for Vitis and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File
OS-Image	*.ub	Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems

Prebuilt files (only on ZIP with prebult content)

Download

Reference Design is only usable with the specified Vivado/SDK/PetaLinux/SDx version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

- TE0726 "Test Board" Reference Design

Design Flow

Reference Design is available with and without prebuilt files. It's recommended to use TE prebuilt files for first lunch.

Trenz Electronic provides a tcl based built environment based on Xilinx Design Flow.

See also:

- AMD Development Tools#XilinxSoftware-BasicUserGuides
- Vivado Projects - TE Reference Design
- Project Delivery.

The Trenz Electronic FPGA Reference Designs are TCL-script based project. Command files for execution will be generated with "_create_win_setup.cmd" on Windows OS and "_create_linux_setup.sh" on Linux OS.

TE Scripts are only needed to generate the vivado project, all other additional steps are optional and can also executed by Xilinx Vivado/SDK GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality

1. Run _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell:

_create_win_setup.cmd/_create_linux_setup.sh

```
-----------------------Set design paths---------------------------
-- Run Design with: _create_win_setup
-- Use Design Path: <absolute project path>
------------------------------------------------------------------
------------------------TE Reference
Design--------------------------
------------------------------------------------------------------
-- (0)  Module selection guide, project creation...prebuilt export...
-- (1)  Create minimum setup of CMD-Files and exit Batch
-- (2)  Create maximum setup of CMD-Files and exit Batch
-- (3)  (internal only) Dev
-- (4)  (internal only) Prod
-- (c)  Go to CMD-File Generation (Manual setup)
-- (d)  Go to Documentation (Web Documentation)
-- (g)  Install Board Files from Xilinx Board Store (beta)
-- (a)  Start design with unsupported Vivado Version (beta)
-- (x)  Exit Batch (nothing is done!)
----
Select (ex.:'0' for module selection guide):
```

2. Press 0 and enter to start "Module Selection Guide"
3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x:\<design name>)
4. Create Project (follow instruction of the product selection guide), settings file will be configured automatically during this process
    - (optional for manual changes) Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd"

Note: Select correct one, see also Vivado Board Part Flow

5. Create hardware description file (.xsa file) for PetaLinux project and export to prebuilt folder

   run on Vivado TCL (Script generates design and export files into "<project folder>\prebuilt\hardware\<short name>")

   ```
   TE::hw_build_design -export_prebuilt
   ```

   Using Vivado GUI is the same, except file export to prebuilt folder.

6. Create and configure your PetaLinux project with exported .xsa-file, see PetaLinux KICKstart
   - use TE Template from "<project folder>\os\petalinux"
   - use exported .xsa file from "<project folder>\prebuilt\hardware\<short name>" . **Note:** HW Export from Vivado GUI creates another path as default workspace.
   - The build images are located in the "<plnx-proj-root>/images/linux" directory
7. Configure the **boot.scr** file as needed, see Distro Boot with Boot.scr
8. Copy PetaLinux build image files to prebuilt folder
   - copy **u-boot.elf**, **image.ub** and **boot.scr** from "<plnx-proj-root>/images/linux" to prebuilt folder

     "<project folder>\prebuilt\os\petalinux\<ddr size>" or "<project folder>\prebuilt\os\petalinux\<short name>"

9. Generate Programming Files with Vitis

   run on Vivado TCL (Script generates applications and bootable files, which are defined in "test_board\sw_lib\apps_list.csv")

   ```
   TE::sw_run_vitis -all
   TE::sw_run_vitis (optional; Start Vitis from Vivado GUI or start with TE Scripts on Vivado TCL)
   ```

   TCL scripts generate also platform project, this must be done manually in case GUI is used. See Vitis

Launch

Programming

Check Module and Carrier TRMs for proper HW configuration before you try any design.

Reference Design is also available with prebuilt files. It's recommended to use TE prebuilt files for first launch.

Xilinx documentation for programming and debugging: Vivado/SDK/SDSoC-Xilinx Software Programming and Debugging

Get prebuilt boot binaries

1. Run _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell
2. Press 0 and enter to start "Module Selection Guide"
    a. Select assembly version
    b. Validate selection
    c. Select Create and open delivery binary folder

> Note: Folder "<project folder>_binaries_<Article Name>" with subfolder "boot_<app name>" for different applications will be generated

QSPI-Boot mode

Option for **Boot.bin** on QSPI Flash and **image.ub** and **boot.scr** on **SD** or **USB**.

1. Connect **JTAG** and power module (TE0726 can be powered via JTAG USB or external)
2. Open Vivado Project with "vivado_open_existing_project_guimode.cmd" or if not created, create with "vivado_create_project_guimode.cmd"
3. Type on Vivado TCL Console: TE::pr_program_flash -swapp u-boot

run on Vivado TCL (Script programs BOOT.bin on QSPI flash)

```
TE::pr_program_flash -swapp u-boot
TE::pr_program_flash -swapp hello_te0820 (optional)
```

> To program with Vitis/Vivado GUI, use special FSBL (fsbl_flash) on setup

4. Copy **image.ub** and **boot.scr** on **SD** or **USB**
    - use files from (<project foler>/_binaries_<Articel Name>)/boot_linux from generated binary folder,see: Get prebuilt boot binaries
    - or use prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
    - Important: Do not copy Boot.bin on SD(is not used see SD note), only other files.
5. Insert **SD**-Card

SD

Xilinx Zynq devices in CLG225 package do not support SD Card boot directly from ROM bootloader. Use QSPI for primary boot and SD for secondary boot (uboot)

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section Programming
2. Connect UART USB (most cases same as JTAG)
3. Insert SD Card with image.ub

> Note: See TRM of the board, which is used.

> Starting with Petalinux version 2020.1, the industry standard "Distro-Boot" boot flow for U-Boot was introduced, which significantly expands the possibilities of the boot process and has the primary goal of making booting much more standardised and predictable.
> The boot options described above describe the common boot processes for this hardware; other boot options are possible.
> For more information see Distro Boot with Boot.scr

4. Power On PCB
    1. Zynq Boot ROM loads FSBL from QSPI into OCM,
    2. FSBL init PS, programs PL using the bitstream and loads U-boot from QSPI into DDR,
    3. U-boot loads Linux (**image.ub**) from SD/QSPI/... into DDR

Linux

1. Open Serial Console (e.g. putty)
    a. Speed: 115200
    b. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

    > Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

2. Linux Console:

```
petalinux login: root
Password: root
```

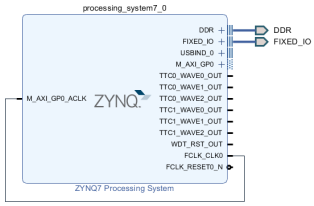
3. You can use Linux shell now.

```
i2cdetect -y -r 0        (check I2C 0 Bus)
dmesg | grep rtc        (RTC check)
udhcpc                                (ETH0 check)
lsusb                                (USB check)
```

4. Option Features
    a. Webserver to get access to Zynq
        - insert IP on web browser to start web interface
    b. init.sh scripts
        - add init.sh script on SD, content will be load automatically on startup (template included in ./misc/SD)

System Design - Vivado

Block Design



Block Design

PS Interfaces

Type	Note
DDR	---
QSPI	MIO
SD1	MIO
I2C1	MIO
UART1	MIO
GPIO	MIO
TTC0..1	EMIO
WDT	EMIO
USB0	MIO, ETH over USB
USB RST	MIO

PS Interfaces

Constrains

Basic module constrains

_i_bitgen_common.xdc

```
#
# Common BITGEN related settings for TE0726
#
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]

set_property BITSTREAM.CONFIG.UNUSEDPIN PULLUP [current_design]
```

Design specific constrain

Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis

Application

Template location: ./sw_lib/sw_apps/

zynq_fsbl

TE modified 2020.2 FSBL

General:

- Modified Files:main.c, fsbl_hooks.h/.c (search for 'TE Mod' on source code)
- Add Files: te_fsbl_hooks.h/.c(for hooks and board)\n\
- General Changes:
    - Display FSBL Banner and Device ID

Module Specific:

- ---

zynq_fsbl_flash

TE modified 2020.2 FSBL

General:

- Modified Files: main.c
- General Changes:
    - Display FSBL Banner
    - Set FSBL Boot Mode to JTAG
    - Disable Memory initialisation

hello_te0726

Hello TE0726 is a Xilinx Hello World example as endless loop instead of one console output.

u-boot

U-Boot.elf is generated with PetaLinux. SDK/HSI is used to generate Boot.bin.

Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart

Config

Start with **petalinux-config** or **petalinux-config --get-hw-description**

Changes:

For 512MB variant:

- No changes

For 64MB variant only:

- CONFIG_SUBSYSTEM_AUTOCONFIG_U__BOOT = y
- CONFIG_SUBSYSTEM_NETBOOT_OFFSET = 0x2000000

For 128MB variant only:

- CONFIG_SUBSYSTEM_AUTOCONFIG_U__BOOT = y
- CONFIG_SUBSYSTEM_NETBOOT_OFFSET = 0x4000000

U-Boot

Start with **petalinux-config -c u-boot**

Changes:

- CONFIG_ENV_IS_NOWHERE=y
- # CONFIG_ENV_IS_IN_SPI_FLASH is not set

Change platform-top.h:

Device Tree

```
/include/ "system-conf.dtsi"
/ {
};


/* USB PHY */

/{
    usb_phy0: usb_phy@0 {
        compatible = "ulpi-phy";
        #phy-cells = <0>;
        reg = <0xe0002000 0x1000>;
        view-port = <0x0170>;
        drv-vbus;
    };
};

&usb0 {
    dr_mode = "host";
    //dr_mode = "peripheral";
    usb-phy = <&usb_phy0>;
};

/* I2C1 */

&i2c1 {
    #address-cells = <1>;
```

```
    #size-cells = <0>;

    i2cmux0: i2cmux@70  {
        compatible = "nxp,pca9544";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x70>;


        i2c1@0 {
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <0>;

            id_eeprom@50 {
                compatible = "atmel,24c32";
                reg = <0x50>;
            };

        };
        i2c1@1 {    // Display Interface Connector
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <1>;
        };
        i2c1@2 {    // HDMI Interface Connector
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <2>;
        };
        i2c1@3 {    // Camera Interface Connector
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <3>;
        };
    };

};
```

Kernel

Start with **petalinux-config -c kernel**

Changes:

- CONFIG_MII=y
- CONFIG_XILINX_GMII2RGMII=y
- CONFIG_USB_USBNET=y
- CONFIG_USB_NET_AX8817X=y
- CONFIG_USB_NET_AX88179_178A=y
- CONFIG_USB_NET_CDCETHER=y
- # CONFIG_USB_NET_CDC_EEM is not set
- CONFIG_USB_NET_CDC_NCM=y
- # CONFIG_USB_NET_HUAWEI_CDC_NCM is not set
- # CONFIG_USB_NET_CDC_MBIM is not set
- # CONFIG_USB_NET_DM9601 is not set
- # CONFIG_USB_NET_SR9700 is not set
- # CONFIG_USB_NET_SR9800 is not set

- # CONFIG_USB_NET_SMSC75XX is not set
- CONFIG_USB_NET_SMSC95XX=y
- # CONFIG_USB_NET_GL620A is not set
- CONFIG_USB_NET_NET1080=y
- # CONFIG_USB_NET_PLUSB is not set
- # CONFIG_USB_NET_MCS7830 is not set
- # CONFIG_USB_NET_RNDIS_HOST is not set
- CONFIG_USB_NET_CDC_SUBSET_ENABLE=y
- CONFIG_USB_NET_CDC_SUBSET=y
- # CONFIG_USB_ALI_M5632 is not set
- # CONFIG_USB_AN2720 is not set
- CONFIG_USB_BELKIN=y
- CONFIG_USB_ARMLINUX=y
- # CONFIG_USB_EPSON2888 is not set
- # CONFIG_USB_KC2190 is not set
- CONFIG_USB_NET_ZAURUS=y
- # CONFIG_USB_NET_CX82310_ETH is not set
- # CONFIG_USB_NET_KALMIA is not set
- # CONFIG_USB_NET_QMI_WWAN is not set
- # CONFIG_USB_NET_INT51X1 is not set
- # CONFIG_USB_SIERRA_NET is not set
- # CONFIG_USB_VL600 is not set
- # CONFIG_USB_NET_CH9200 is not set
- # CONFIG_USB_NET_AQC111 is not set
- CONFIG_USBIP_CORE=y
- # CONFIG_USBIP_VHCI_HCD is not set
- # CONFIG_USBIP_HOST is not set
- # CONFIG_USBIP_VUDC is not set
- # CONFIG_USBIP_DEBUG is not set

Change linux-xlnx_%.bbappend:

```
FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}:"

SRC_URI += "file://devtool-fragment.cfg \
            file://0001-QSPI-s25fl127_8-2020_2.patch \
            "
```

- Add 0001-QSPI-s25fl127_8-2020_2.patch to "<project folder>\project-spec\meta-user\recipes-kernel\linux\linux-xlnx\"

Rootfs

Start with **petalinux-config -c rootfs**

Changes:

- CONFIG_i2c-tools=y
- CONFIG_busybox-httpd=y (for web server app)
- CONFIG_packagegroup-petalinux-utils = y
- CONFIG_util-linux-umount=y
- CONFIG_util-linux-mount=y

Applications

startup

Script App to load init.sh from SD Card if available.

See: \os\petalinux\project-spec\meta-user\recipes-apps\startup\files

webfwu

Webserver application accemble for Zynq access. Need busybox-httpd

Additional Software

No additional software is needed

Appx. A: Change History and Legal Notices

Document Change History

To get content of older revision got to "Change History" of this page and select older document revision number.

Date	Document Revision	Authors	Description
Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot	• 2020.2 release

resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.

resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.

resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.

atlassian . confluence.core. ContentEntityObject] [interface com. atlassian.user. User, class java. lang. String, class com. atlassian . confluence.core. ContentEntityObject]	atlassian . confluence.core. ContentEntityObject] [interface com. atlassian.user. User, class java. lang. String, class com. atlassian . confluence.core. ContentEntityObject]	atlassian . confluence.core. ContentEntityObject] [interface com. atlassian.user. User, class java. lang. String, class com. atlassian . confluence.core. ContentEntityObject]	
2020-06-23	v.17	John Hartfiel	• typo
2020-04-08	v.16	John Hartfiel	• Design update • Programming issue note
2020-03-25	v.14	John Hartfiel	• script update
2020-02-19	v.13	Mohsen Chamanbaz	• 2019.2 release • docu update
2019-12-13	v.12	John Hartfiel	• 2018.3 release

2018-07-13	v.11	John Hartfiel	• 2018.2 release
2018-05-17	v.9	John Hartfiel	• bugfix design for 128MB variant
2018-03-20	v.8	John Hartfiel	• Link update • remove typo
2018-02-16	v.6	John Hartfiel	• Design update
2018-02-09	v.5	John Hartfiel	• 2017.4 release
--	all	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to	--

invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.

		Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	

Document change history.

Legal Notices

Data Privacy

Please also note our data protection declaration at https://www.trenz-electronic.de/en/Data-protection-Privacy

Document Warranty

The material contained in this document is provided “as is” and is subject to being changed at any time without notice. Trenz Electronic does not warrant the accuracy and completeness of the materials in this document. Further, to the maximum extent permitted by applicable law, Trenz Electronic disclaims all warranties, either express or implied, with regard to this document and any information contained herein, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non infringement of intellectual property. Trenz Electronic shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

Limitation of Liability

In no event will Trenz Electronic, its suppliers, or other third parties mentioned in this document be liable for any damages whatsoever (including, without limitation, those resulting from lost profits, lost data or business interruption) arising out of the use, inability to use, or the results of use of this document, any documents linked to this document, or the materials or information contained at any or all such documents. If your use of the materials or information from this document results in the need for servicing, repair or correction of equipment or data, you assume all costs thereof.

Copyright Notice

Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

Environmental Protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

REACH, RoHS and WEEE

REACH

Trenz Electronic is a manufacturer and a distributor of electronic products. It is therefore a so called downstream user in the sense of REACH. The products we supply to you are solely non-chemical products (goods). Moreover and under normal and reasonably foreseeable circumstances of application, the goods supplied to you shall not release any substance. For that, Trenz Electronic is obliged to neither register nor to provide safety data sheet. According to present knowledge and to best of our knowledge, no SVHC (Substances of Very High Concern) on the Candidate List are contained in our products. Furthermore, we will immediately and unsolicited inform our customers in compliance with REACH - Article 33 if any substance present in our goods (above a concentration of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA).

RoHS

Trenz Electronic GmbH herewith declares that all its products are developed, manufactured and distributed RoHS compliant.

WEEE

Information for users within the European Union in accordance with Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE).

Users of electrical and electronic equipment in private households are required not to dispose of waste electrical and electronic equipment as unsorted municipal waste and to collect such waste electrical and electronic equipment separately. By the 13 August 2005, Member States shall have ensured that systems are set up allowing final holders and distributors to return waste electrical and electronic equipment at least free of charge. Member States shall ensure the availability and accessibility of the necessary collection facilities. Separate collection is the precondition to ensure specific treatment and recycling of waste electrical and electronic equipment and is necessary to achieve the chosen level of protection of human health and the environment in the European Union. Consumers have to actively contribute to the success of such collection and the return of waste electrical and electronic equipment. Presence of hazardous substances in electrical and electronic equipment results in potential effects on the environment and human health. The symbol consisting of the crossed-out wheeled bin indicates separate collection for waste electrical and electronic equipment.

Trenz Electronic is registered under WEEE-Reg.-Nr. DE97922676.

Error rendering macro 'page-info'

Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]