# TEM0005 Webserver Demo1



## Overview

This demo is a Webserver which utilizes SmartFusion2 SoC ARM Cortex-M3, Ethernet, USB / COM-port, Real Time Clock and the on-board LEDs.

The demo is offered in two variants, one which is stored into the embedded non-volatile memory (eNVM) and the seconde one which stored to the external DDR3/L SDRAM memory and therefore volatile.

Refer to http://trenz.org/tem0005-info for the current online version of this manual and other available documentation.

## Key Features

- Libero 12.4 (FPGA IDE)
- SoftConsole 6.2 (Software IDE)
- FreeRTOS V7.0.1 (Free real time operating system)
- lwIP (lightweight IP)
- ETH
- UART
- DDR
- eNVM
- User LEDs
- User button
- Real Time Clock

## Revision History

| Date | Libero | Project Built | Authors | Description |
|---|---|---|---|---|
| 2021-06-02 | 12.4 | TEM0005-01-010C_ReferenceDesigns01_Lib12.4-SC6.2_20210602-1940.zip | Kilian Jahn | • Demo Webserver ported from TEM0002 /Libero 11.8<br>• Adapted the User Interface<br>• Added "Hello World" (into SoftConsole workspace)<br>• Added "SoC Erase" (into SoftConsole workspace) |
| 2018-02-26 | 11.8 | Smartberry_Webserver_Demo.zip | -- | • Initial release |

**Design Revision History**

## Release Notes and Know Issues

| Issues | Description | Workaround/Solution | To be fixed version |
|---|---|---|---|
| Webserver Demo<br>▪ Google search page | Search fails | Unknown | Unknown |

**Known Issues**

# Requirements

## Software

| Software | Version | Note |
|---|---|---|
| Windows 10 | 2004 / 19041 | |
| Libero Release | 12.4 | |
| SoftConsole | 6.2 | Included in the Libero installation |
| Microsemi Flash Pro 5 module driver | 2.10.0.0 | Utilize onboard programmer and USB / comport connection. Included in the Libero installation |
| FTDI Driver for the TEM0005 module | 2.12.28.0 | |
| UART / COM-port terminal | | Capturing the modules messages |
| Web browser | | Optional for the Demo Webserver, an ordinary Web browser (supporting MS-HTML > 6.0) |

**Software**

## Hardware

Design supports following modules:

| Module Model | PCB Revision Support | DDR | embedded SRAM | Embedded Flash | Notes |
|---|---|---|---|---|---|
| TEM0005-01-010C | REV02 | 2 GBit / 256 MB | 64 kB | 256 kB | NA |

**Hardware Modules**

Additional hardware Requirements:

| Additional Hardware | Notes |
|---|---|
| Demo host computer | Demo was created and tested on windows |
| Carrier TEMB0005 | |
| Micro USB to USB Type A Cable | Power supply, JTAG: Programming the board, UART: Communication Interface to the board. |

| | |
|---|---|
| ETH cable | Hardware for the Demo Webserver. |
| Router / LAN to USB bridge | Hardware for the Demo Webserver. |

**Additional Hardware**

# Content

Content of the zip archive "TEM0005-01-010C_ReferenceDesigns01_LibXY.Z-SCX.Y_Date-Time":

## Design Sources

| Type | Location | Notes |
|---|---|---|
| Libero | <zip archive><br>/ Libero-RefDes_vXY | Libero Project containing the modules<br>Hardware Reference Design |
| FlashPro Express | <zip archive><br>/ FlashPro_Express<br>/ Libero-RefDes-v01<br><br>/SoC_HW_RefDes_v01.pro | FlashPro Express Hardware Design programming file |
| SoftConsole | <zip archive><br>/ SC-Workspace<br>/ Smartberry_Hello_World_X.y<br>/ Smartberry_Webserver_X.y<br>/ Smartberry_Webserver_DDR_X.y | SoftConsole Workspace contains the Software Projects :<br>▪ Hello_World<br>and two variants of the Demo :<br>▪ Smartberry_Webserver |
| SoftConsole | <zip archive><br>/ Softconsole-X.y-Workspace<br>/ microsemi-smartfusion2-smartberry-ddr .cfg | Board configuration file, needed to<br><br>debug / run applications |

**Design sources**

# Download

The Trenz Electronic Reference Designs and Demos are usable with the specified Microsemi Libero / SoftConsole version. Usage of a different Microsemi Libero / SoftConsole software versions is not recommended.

Reference Designs / Demos are available via the following link:

- TEM0005 Reference Designs/Demos

The download is a ZIP compressed archive. Extract the archive before usage.

# Design Flow

The Hardware and Software Reference / Demo -Designs Projects are available as a prebuild zip archive. The archive contains a **Libero Hardware Project** and a **SoftConsole Workspace** folder, they were created and tested in windows environment.

This SoftConsole Workspace contains the Software Projects Webserver, Hello World and SoC erase flash. The **board configuration file** "microsemi-smartfusion2-smartberry-ddr.cfg" which is required for the usage of the Software projects via the IDE SoftConsole.

# Launch

Executing a Reference / Demo Design on a module requires the powering of it and a JTAG or UART Connection for Programming and Communication. Often the programming is a two fold process, where the first programming configures the FPGA and the second programming flashes Software code to be executed inside the FPGA / ARM processor.

## Connecting

Connect the carriers micro USB connector to your host pc and power it via barrel connector (5 V, plus pole inside), this powers the module and a simultaneous JTAG and UART connection is possible.

Only necessary for running the Demo Webserver:
The demo is configured to establish a network connection via the DHCP protocol, therefore, if a a free router port is used, no further port setup is required.
If a "direct Ethernet Connection" between Host PC and module is used, the user must know how to setup this connection type. Further down in this chapter is explained how to setup the Demo Webserver and recompile it, so that it uses a static IP.

## Driver check

When the module is connected via USB cable to your demo host computer, in the Windows Device Manager appear the following tree board driver related devices:

In section **Ports (COM & LPT):**

- FlashPro5 Port (ComX)

In section **Universal Serial Bus controllers:**

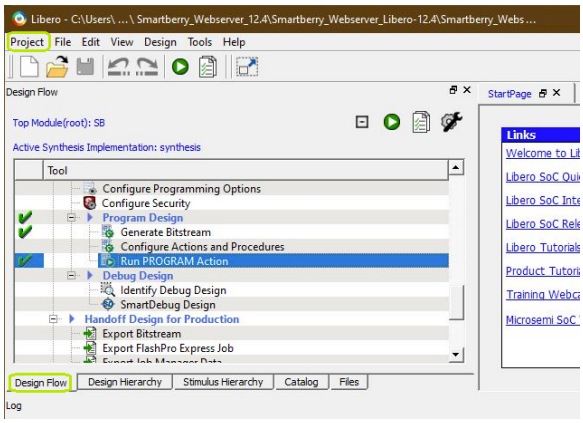
- USB FP5 Serial Converter A
- USB FP5 Serial Converter B

The Device Manager is accessible via "Right mouse click context menu" from the Windows Start Menu Button. When these devices are not visible, the driver installation through libero could be faulty.

## Programming the Hardware design

### Via Libero

Programming of the Hardware reference Design requires to open the FPGA Design IDE Libero.




**Libero GUI "Run PRGORAMM ACTION"**

The Hardware Reference Design can be opened via "Project > Open Project" in the top right corner of Libero (picture above - upper green rectangle). A file dialogue opens, point the dialogue along the extracted download to the folder containing the Hardware Reference Design.

*Disk :\ Path-to-the-demo-archive \ Extracted ZIP-archive \ Libero-RefDesign_vXY*

Double left mouse click onto the project file "Referenz-Design_XY .prjx" to open it. The board is automatically selected and setup to be flashed by Libero.

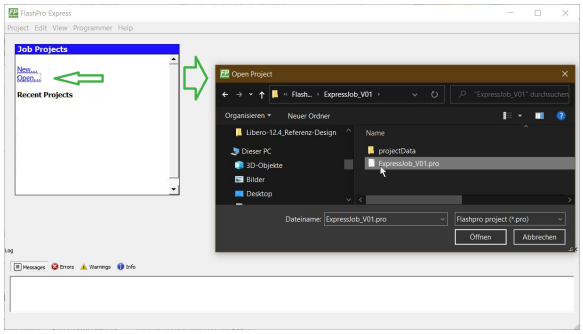
In the upper left section of Libero, select the tab "Design Flow" (picture above - lover green rectangle) and flash it to the board via "Program Design > and double left mouse click onto "Run PROGRAM Action" (picture above - row with blue background).

Warnings should not affect the functionality of a Reference / Demo -Design.

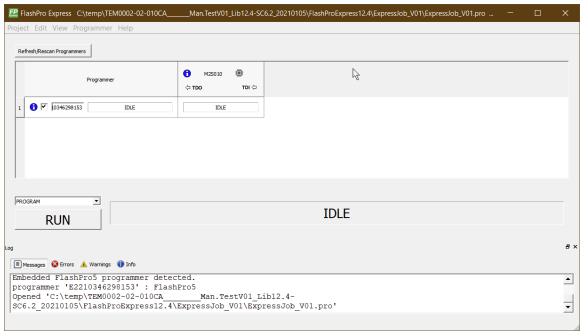
## Via FlashPro Express

Open "FlashPro Express", to the left, in section "Job Project", click on "**Open...**" .

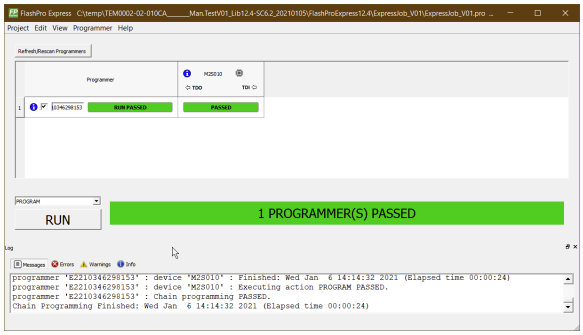
In the appearing file dialogue, point to the projects "Programming job file (.pro-file)" and confirm.




The programm windows changes through opening the file. Now click "Refresh/Rescan Programmers", wait untill the scan is finished.




Click "RUN" to programm the module.




When the programming is done, you can close FlashPro Express.

# UART connection

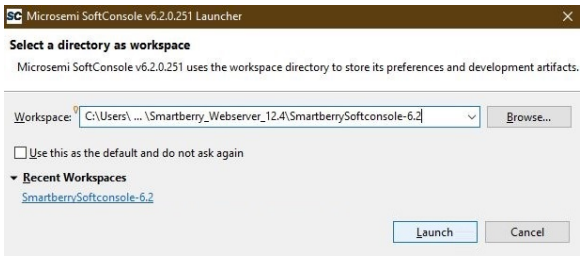
Before flashing any Software Project to the module, open a comport terminal to the boards comport, so that it's messages can be captured.

# Programming a Software project

Open SoftConsole and press "Browse..." near the right edge. A file dialogue opens, point the dialogue along the extracted download to the folder containing the SoftConsole Workspace.

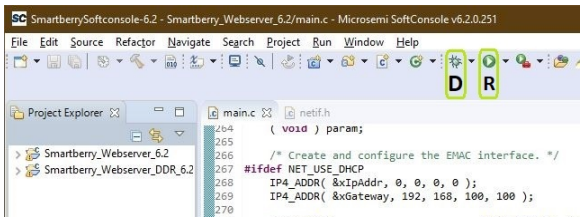
*Disk :\ Path-to-the-demo-archive \ Extracted ZIP-archive \ SC-Workspace *

Confirm your selection by pressing "Ok" , the dialogue closes, and open The SoftConsole by pressing "Launch"




**SoftConsole "Select the Workspace"**

Subsequently the program opens and shows the software project's which are contained inside the workspace. The projects and their files can be accessed in the section "Project Explorer".




**SoftConsole GUI**

To simply run a Project, press the triangle right to the button marked with a "R" in the picture above and select a demo.

Pressing the triangle next to the button marked with "D" let you select which variant to be executed in debug mode.

Debug controls - Resume - Pause - Stop

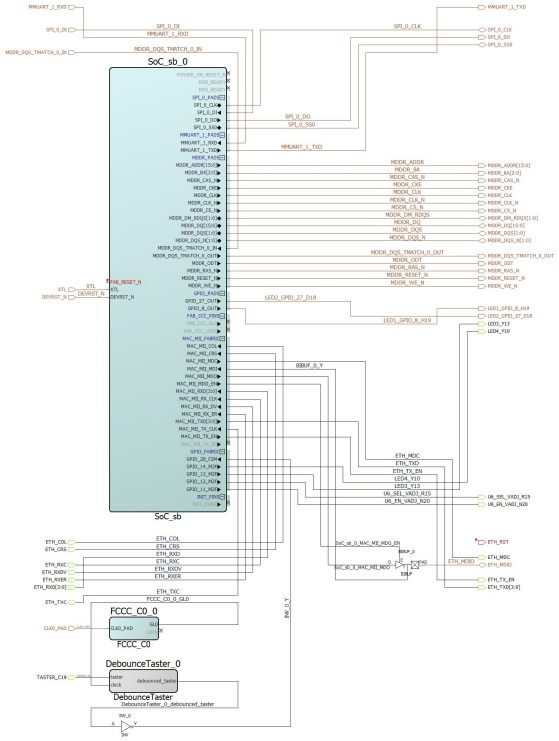
**SoftConsole "Debug controlls"**

Switch between Debug and Run perspective (upper right corner program window)

**SoftConsole "Switch GUI layout"**

# System Design - Libero

## Smart Design




**Block Design**

# Constrains

## IO constrains

**user.pdc**

```
# Microsemi I/O Physical Design Constraints file

# User I/O Constraints file

# Version: v12.4 12.900.0.16

# Family: SmartFusion2 , Die: M2S010 , Package: 400 VF

# Date generated: Wed May 26 09:19:42 2021

#
# User Locked I/O Bank Settings
#

#
# Unlocked I/O Bank Settings
# The I/O Bank Settings can be locked by directly editing this file
# or by making changes in the I/O Attribute Editor
#

#
# User Locked I/O settings
#

set_io CLK0_PAD       \
    -pinname Y12      \
    -fixed yes        \
    -DIRECTION INPUT


set_io ETH_COL        \
    -pinname D2       \
    -fixed yes        \
    -DIRECTION INPUT


set_io ETH_CRS        \
    -pinname E3       \
    -fixed yes        \
    -DIRECTION INPUT


set_io ETH_MDC         \
    -pinname F3        \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io ETH_MDIO       \
    -pinname F4       \
    -fixed yes        \
    -DIRECTION INOUT


set_io ETH_RST         \
```

```
    -pinname H6        \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io ETH_RXC        \
    -pinname G1       \
    -fixed yes        \
    -DIRECTION INPUT


set_io ETH_RXDV       \
    -pinname E1       \
    -fixed yes        \
    -DIRECTION INPUT


set_io {ETH_RXD[0]}   \
    -pinname J7       \
    -fixed yes        \
    -DIRECTION INPUT


set_io {ETH_RXD[1]}   \
    -pinname F1       \
    -fixed yes        \
    -DIRECTION INPUT


set_io {ETH_RXD[2]}   \
    -pinname H5       \
    -fixed yes        \
    -DIRECTION INPUT


set_io {ETH_RXD[3]}   \
    -pinname H4       \
    -fixed yes        \
    -DIRECTION INPUT


set_io ETH_RXER       \
    -pinname E2       \
    -fixed yes        \
    -DIRECTION INPUT


set_io ETH_TXC        \
    -pinname G2       \
    -fixed yes        \
    -DIRECTION INPUT


set_io {ETH_TXD[0]}    \
    -pinname G4        \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io {ETH_TXD[1]}    \
    -pinname F5        \
```

```
    -fixed yes         \
    -DIRECTION OUTPUT


set_io {ETH_TXD[2]}    \
    -pinname G6        \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io {ETH_TXD[3]}    \
    -pinname F7        \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io ETH_TX_EN       \
    -pinname G3        \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io LED3_Y13        \
    -pinname Y13       \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io LED4_Y10        \
    -pinname Y10       \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io TASTER_C19     \
    -pinname C19      \
    -fixed yes        \
    -DIRECTION INPUT


set_io U6_EN_VADJ_N20  \
    -pinname N20       \
    -fixed yes         \
    -DIRECTION OUTPUT


set_io U6_SEL_VADJ_R15  \
    -pinname R15        \
    -fixed yes          \
    -DIRECTION OUTPUT



#
# Dedicated Peripheral I/O Settings
#


#
# Unlocked I/O settings
# The I/Os in this section are unplaced or placed but are not locked
```

```
# the other listed attributes have been applied
#


#
#Ports using Dedicated Pins

#

set_io DEVRST_N       \
    -pinname U17      \
    -DIRECTION INPUT


set_io XTL            \
    -pinname Y18      \
    -DIRECTION INPUT
```

## Clock constrains

**user.sdc**

```
create_clock -name {XTAL_12MHz} -period 83.3333 -waveform {0 41.6667 } [
get_ports { XTL } ]
create_clock -name {Clk0pad_30MHz_Y12} -period 33.3333 -waveform {0
16.6667 } [ get_ports { CLK0_PAD } ]
create_clock -name {SoC_sb_0___SoC_sb_MSS_0___FIC_2_APB_M_PCLK} -period 10
-waveform {0 5 } [ get_nets { SoC_sb_0/SoC_sb_MSS_0/FIC_2_APB_M_PCLK } ]
create_clock -name {SoC_sb_0___FABOSC_0___N_RCOSC_25_50MHZ_CLKOUT} -period
10 -waveform {0 5 } [ get_nets { SoC_sb_0/FABOSC_0/N_RCOSC_25_50MHZ_CLKOUT
} ]
create_clock -name {MAC_RXC_TXC} -period 40 -waveform {0 20 } [ get_ports
{ ETH_RXC ETH_TXC } ]
create_clock -name {SoC_sb_0___CCC_0___GL0} -period 10000 -waveform {0
5000 } [ get_nets { SoC_sb_0/CCC_0/GL0_net } ]
create_clock -name {FCCC_C0_FCCC_C0_0_FCCC___GL0_net_inferred_clock} -
period 10000 -waveform {0 5000 } [ get_pins { FCCC_C0_0/FCCC_C0_0/CCC_INST
/GL0 } ]
```

# Software Design - SoftConsole

## Applications

### Webserver

The demo projects "Webserver_DHCP" and "Webserver_DDR_STATIC" are identical variants of the demo, they only differ in their memory location:

- Webserver_DHCP - Application code is stored to the FPGA's embedded non-volatile memory (e NVM)
- Webserver_DDR_STATIC - Application code is stored to the FPGA's external volatile memory ( DDR3 RAM) and lost during power down or reset

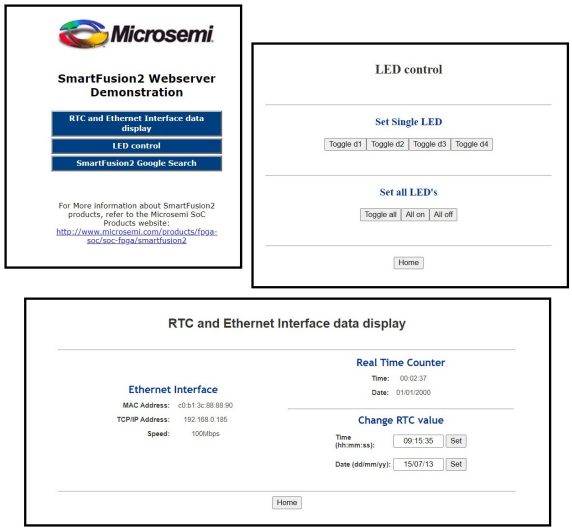
Initialization UART messages after programming:

```
**********************************************************************
##  Welcome to SmartFusion2 Webserver TCP Demo (Flash / SC-6.2)  ##
**********************************************************************
          Initializing the MAC and getting IP Address
**********************************************************************
                      DHCP Service is used
**********************************************************************

 Use the following IP address to browse the Webserver
 Requested IP address : 192.168.0.185
```

**COM-port Terminal Webserver "Welcome / IP -message"**

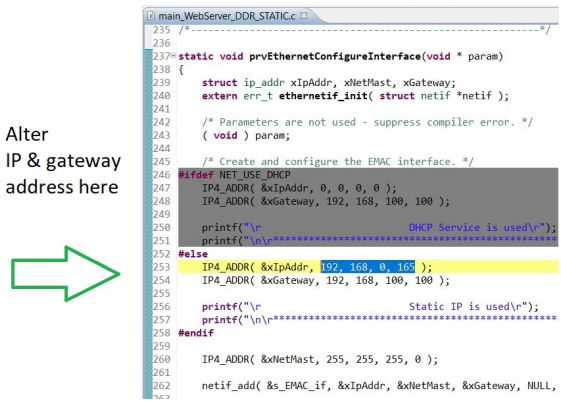
Webpage in the browser:




**COM-port Terminal Webserver "Welcome / IP -message"**

## Static IP configuration

To change IP and Gateway Address of the project Webserver_DDR_STATIC, open the project in SoftConsole. Open the file main_WebServer_DDR_Static and scroll to line 253. Change the addresses to your needs.

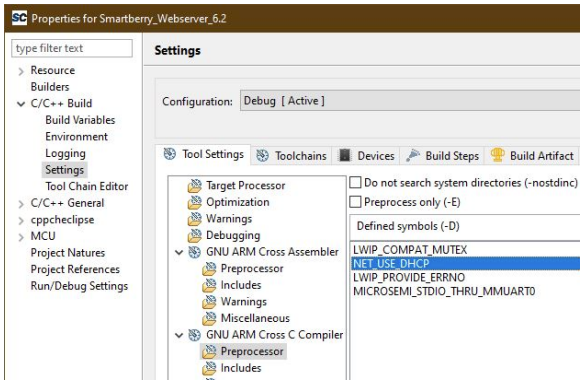


**SoftConsole "main.c - Set IP"**

### Disengage the DHCP service

To disengaging the DHCP mode one has to setup up an IP and Gateway Address in the code, this is discribed in the chapter above. Alternativly, the demo hosts IP Address can be changed.

Furthermore the corresponding compiler flag needs to be deleted in the project setting. To do so, in the "Project Explorer" tab, right mouse click onto the project and select Properties in the appearing menu.




**SoftConsole "Static IP- Change Defines""**

In the left section of the properties window select "C/C++ Build > Settings" in the right section select the tab "Tool Settings > GNU ARM Cross C Compiler > Preprocessor" under "Defined symbols (-D)" delete the compiler flag "NET_USE_DHCP" and press "Apply". Confirm the following dialogue and press "Cancel".

Lastly, the project needs to be recompiled. In the top menu of the SoftConsole select "Project > Build ALL / Build Project".

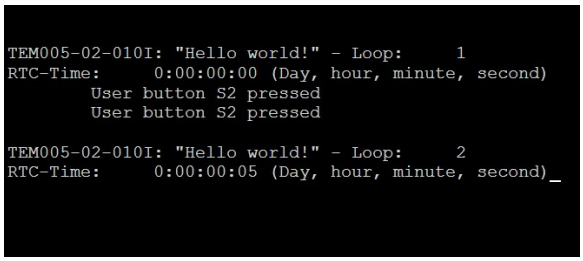
Warnings should not affect the demo and can be ignored.

## HelloWorld

Hello World example as endless loop. Each loop contains the updated current loop and RTC/system time.

The user LED's are continuously blinking. The user buttons responds with a message at any time.

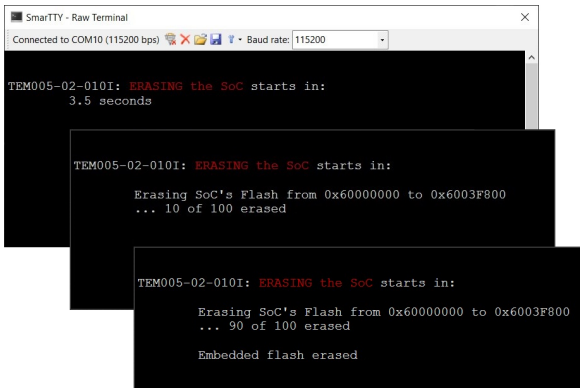
UART output:




**COM-port Terminal "Hello World loop"**

## SoC-erase

The SoC's embedded non-volatile memory (eNVM) can only be erased from within.




**COM-port Terminal "SoC erase flash"**

# 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'**<br>Ambiguous method overloading for method jdk.proxy27.9.$Proxy4022#haConteLevelPe | **Error rendering macro 'page-info'**<br>Ambiguous method overloading for method jdk.proxy27.9.$Proxy4022#haConteLevelPe | **Error rendering macro 'page-info'**<br>Ambiguous method overloading for method jdk.proxy27.9.$Proxy4022#haConteLevelPe | • Link updates<br>• table of content update |

rmission . Cannot resolve which method to invoke for [null, class java. lang. String, class com. atlassian . confluen ce. pages. Page] due to overlapp ing prototyp es between : [interfac e com. atlassian . confluen ce.user. Conflue nceUser , class java. lang.

rmission . Cannot resolve which method to invoke for [null, class java. lang. String, class com. atlassian . confluen ce. pages. Page] due to overlapp ing prototyp es between : [interfac e com. atlassian . confluen ce.user. Conflue nceUser , class java. lang.

rmission . Cannot resolve which method to invoke for [null, class java. lang. String, class com. atlassian . confluen ce. pages. Page] due to overlapp ing prototyp es between : [interfac e com. atlassian . confluen ce.user. Conflue nceUser , class java. lang.

| | | | |
|---|---|---|---|
| String, class com. atlassian . confluen ce.core. Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject] | String, class com. atlassian . confluen ce.core. Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject] | String, class com. atlassian . confluen ce.core. Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject] | |
| 20-11-24 | v.41 | Kilian Jahn | • Libero12.4 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.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.

| | | | |
|---|---|---|---|
| | | atlassian . confluence.core. ContentEntityObject] | |

**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]