

TEM0002 Webserver Demo1

- 1 Overview
 - 1.1 Key Features
 - 1.2 Revision History
 - 1.3 Release Notes and Know Issues
 - 1.4 Requirements

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 second one which is stored to the external DDR3/L SDRAM memory and therefore volatile.

Refer to <http://renz.org/tem0002-info> for the current online version of this manual and other available documentation.

Key Features

- 4 System Design - Libero
 - Libero 12.4 Smart IDE (FPGA IDE)
 - SoftConsole 6.2 (Software IDE)
- 5 Software Design - SoftConsole
 - FreeRTOS V7.9.1 (Free real time operating system)
 - lwIP 1.4.1 (lightweight IP)
 - ETH
 - 5.1.1 Application
 - 5.1.1.1 Demo - Webserver_...
 - 5.1.1.1.1 Static IP configuration
 - 5.1.2 Reference Design - HelloWorld_...
 - UART
 - DDR
 - eNVM
- 6 Appendix
 - User License
 - Real Time Clock
 - 6.1 Legal Notices
 - 6.2 Data Privacy
 - 6.3 Document Warranty
 - 6.4 Intellectual Liability
 - 6.5 Copyright Notice
 - 6.6 Technology Licenses
 - 6.7 Environmental Protection
 - 6.8 REACH, RoHS and WEEE

Revision History

Date	Libero	Project Built	Authors	Description
2020-11-23	12.4	TEM0002-SmartBerry_Webserver-Demo_Libero-12.4_20201123-1511 .zip	Kilian Jahn	• Added "Hello World" (into SoftConsole workspace) • Demo Webserver ported from Libero 11.8 • Improved its User Interface
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 ▪ 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 TEM0002 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	Board Part Short Name	PCB Revision Support	DDR	embedded SDRAM	Embedded Flash	Notes
TEM0002-01-010C	SmartBerry	REV01	1 GBit / 128 MB	64 kB	256 kB	NA

Hardware Modules

Additional hardware Requirements:

Additional Hardware	Notes
Demo host computer	Demo was created and tested on windows
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 "TEM0002-SmartBerry_Webserver-Demo_Libero-X.y_Datum-Time":

Design Sources

Type	Location	Notes
Libero	<zip archive> / Libero-X.y_Referenz- Design_XY	Libero Project containing the modules Hardware Reference Design
SoftConsole	<zip archive> / Softconsole-X.y- Workspace / Smartberry_Hello_World_X.y / Smartberry_Webserver_X.y / Smartberry_Webserver_DDR_X .y	SoftConsole Workspace contains the Software Projects : ▪ Hello_World and two variants of the Demo : ▪ Smartberry_Webserver
SoftConsole	<zip archive> / Softconsole-X.y- Workspace / microsemi- smartfusion2-smartberry-ddr .cfg	Board configuration file, needed to 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:

- [TEM0002_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 Project Hello World and the Demo Webserver, the demo is offered in two variants. The **board configuration file** "microsemi-smartfusion2-smartberry-ddr.cfg" 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 modules micro USB connector to your host pc, this enables the powering of the module and a simultaneous JTAG and UART connection .

Only necessary for running the Demo Webserver:

The demo is configured to establish a network connection via the DHCP protocol, therefore, if 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)

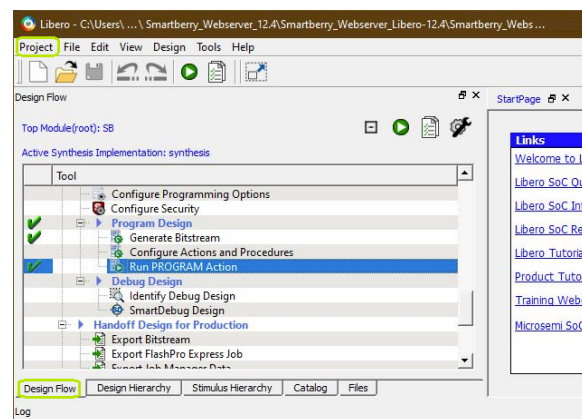
In section **Universal Serial Bus controllers**:

- 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

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-X.y_Reference-Design\

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 - lower 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.

UART connection

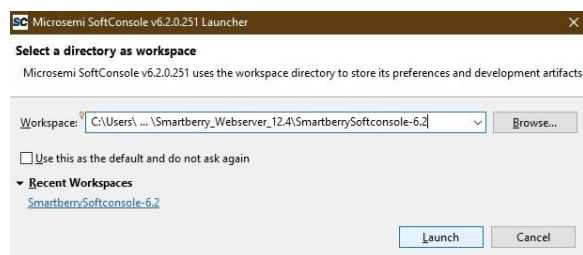
Before flashing any Software Project to the module, open a comport terminal to the board's comport, so that its messages can be captured.

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.

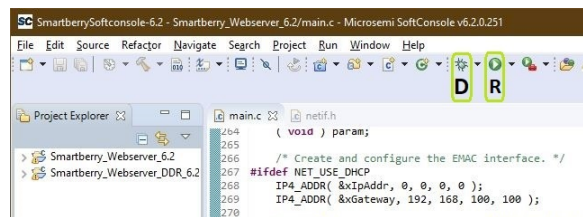
Disk : \ Path-to-the-Demo-archive \ Extracted ZIP-archive \ Softconsole-X.y-Workspace \

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 who are contained inside the workspace to the left, under "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 variant of the 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 controls"

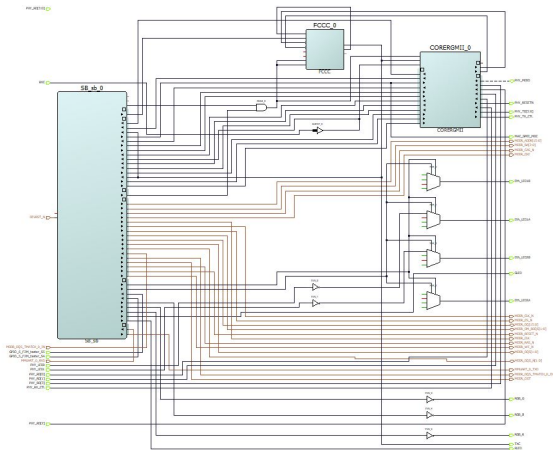
Switch between Debug and Run perspective (upper right corner program window)



SoftConsole "Switch GUI layout"

System Design - Libero

Smart Design



Block Design

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: Mon Nov 16 11:11:16 2020

#

# 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 Eth_LED1A \  
-pinname Y10 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io Eth_LED1B \  
-pinname U12 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io Eth_LED2A \  
-pinname V14 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io Eth_LED2B \  
-pinname U14 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io GLED \  
-pinname G17 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io GPIO_5_F2M_taster_S4 \  
-pinname E17 \  
-fixed yes \  
-DIRECTION INPUT
```

```
set_io GPIO_6_F2M_taster_S5 \  
-pinname E16 \  
-fixed yes \  
-DIRECTION INPUT
```

```
set_io MAC_GMII_MDC \  
-pinname N1 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION OUTPUT
```

```
set_io PHY_LED0 \  
-pinname U11 \  
-fixed yes \  
-DIRECTION INPUT
```

```
set_io PHY_LED1 \  
-pinname T14 \  
-fixed yes \  
-DIRECTION INPUT
```

```
set_io PHY_MDIO \  
-pinname N2 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION INOUT
```

```
set_io {PHY_RD[0]} \  
-pinname K5 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION INPUT
```

```
set_io {PHY_RD[1]} \  
-pinname H1 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION INPUT
```

```
set_io {PHY_RD[2]} \  
-pinname H2 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION INPUT
```

```
set_io {PHY_RD[3]} \  
-pinname J4 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION INPUT
```

```
set_io PHY_RESETN \  
-pinname R13 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io PHY_RX_CTL \  
-pinname K1 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION INPUT
```

```
set_io {PHY_TD[0]} \  
-pinname L1 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION OUTPUT
```

```
set_io {PHY_TD[1]} \  
-pinname M2 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION OUTPUT
```

-DIRECTION OUTPUT

```
set_io {PHY_TD[2]} \  
-pinname M1 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION OUTPUT
```

```
set_io {PHY_TD[3]} \  
-pinname M3 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION OUTPUT
```

```
set_io PHY_TX_CTL \  
-pinname K3 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION OUTPUT
```

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

```
set_io RGB_G \  
-pinname F6 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io RGB_R \  
-pinname H5 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io RLED \  
-pinname G16 \  
-fixed yes \  
-DIRECTION OUTPUT
```

```
set_io RXC \  
-pinname J2 \  
-fixed yes \  
-iostd LVCMOS15 \  
-DIRECTION INPUT
```

```
set_io TXC \  
-pinname K7 \  
-fixed yes \  
-iostd LVCMOS15 \  
-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

```

Software Design - SoftConsole

Application

Demo - Webserver_...

The demo projects "Smartberry_Webserver_X.y" and "Smartberry_Webserver_DDR_X.y" are identical variants of the demo, they only differ in their memory location:

- Smartberry_Webserver_X.y - Application code is stored to the FPGA's embedded non-volatile memory (eNVM)
- Smartberry_Webserver_DDR_X.y - Application code is stored to the FPGA's external volatile memory (DDR3/L SDRAM) and lost during power down

UART output:

```

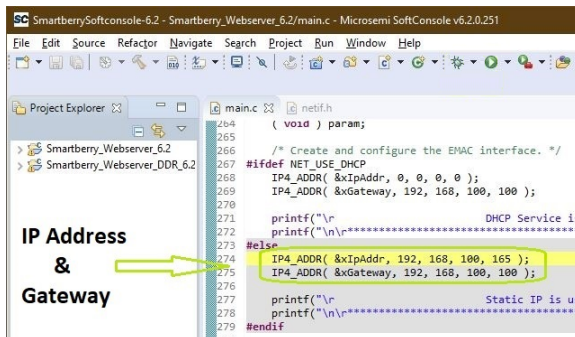
*****
##      Welcome to SmartFusion2 Webserver TCP Demo (Flash / SC-6.2)      ##
*****
      Initializing the MAC and getting IP Address
*****
      DHCP Service is used
*****

Use the following IP address to browse the Webserver
Requested IP address : 192.168.150.127

```

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

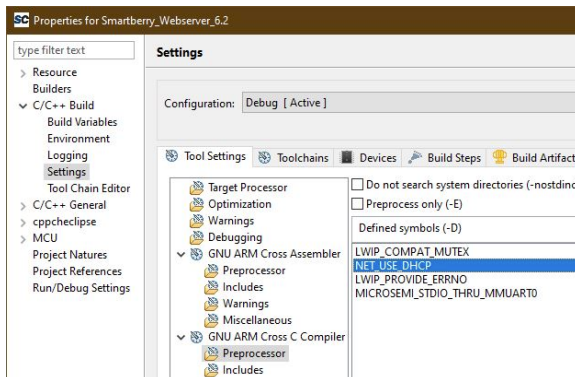
Static IP configuration



SoftConsole "main.c - Set IP"

To disengaging the DHCP mode one has to setup up an IP and Gateway Address in the code unit "main.c" roughly at line 270. Alternatively, 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. can be ignored.

Reference Design - HelloWorld...

Hello World example as endless loop instead of one console output. Each loop lights up each LED. The user buttons responds with a message at any time.

UART output:

```
TEM0002: "Hello world!" - Loop: 1
RTC-Time: 0: 0: 0: 0
Cycle - LED - red
Cycle - LED - green
Cycle - RGB LED - red
Cycle - RGB LED - green
Cycle - RGB LED - blue
Cycle - Eth LED - orange
Cycle - Eth LED - green
--- TEM0002: User button S4 ---
--- TEM0002: User button S5 ---
```

COM-port Terminal "Hello World loop"

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 overload ing for method jdk. proxy24	Error rendering macro 'page-info' Ambiguous us method overload ing for method jdk. proxy24	Error rendering macro 'page-info' Ambiguous us method overload ing for method jdk. proxy24	Download Link update

1.\$Proxy
3496#ha
sConten
tLevelPe
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

1.\$Proxy
3496#ha
sConten
tLevelPe
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

1.\$Proxy
3496#ha
sConten
tLevelPe
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]	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]	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]	
2021-01-29	v.45	John Hartfiel	- Link updates - table of content update
2020-11-24	v.41	Kilian Jahn	Libero12.4 release
--	all		--

**Error
renderi
ng
macro
'page-
info'**

Ambiguo
us
method
overload
ing for
method
jdk.
proxy24
1.\$Proxy
3496#ha
sConten
tLevelPe
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]	
--	--	---	--

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

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Trenz Electronic.

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.
proxy241.\$Proxy3496#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]