BECKHOFF New Automation Technology

Manual | EN

TwinCAT/BSD

Operating system

12/12/2022 | Version: 1.3

Table of contents

1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.1 Symbol explanation

The following warnings are used in the documentation. Read and follow the warnings.

Warnings relating to damage to property or the environment:

NOTE
There is a potential hazard to the environment and equipment.

Notes showing further information or tips:

This notice provides important information that will be of assistance in dealing with the product or software. There is no immediate danger to product, people or environment.

1.2 Documentation issue status

Version	Modifications
1.0	First version
1.1	Chapter TwinCAT/BSD, network settings, Package Server, configuration, TwinCAT 3, recovery options and error handling and diagnostics revised
1.2	Chapter System update, System information, Beckhoff Device Manager and Synchronize time with NTP added. Chapter TwinCAT revised.
1.3	Chapter TwinCAT/BSD Hypervisor added.

2 Setup

2.1 Create bootable USB stick

Before you can install TwinCAT/BSD on a Industrial PC, you must create a bootable USB stick and upload the current image to the USB stick.

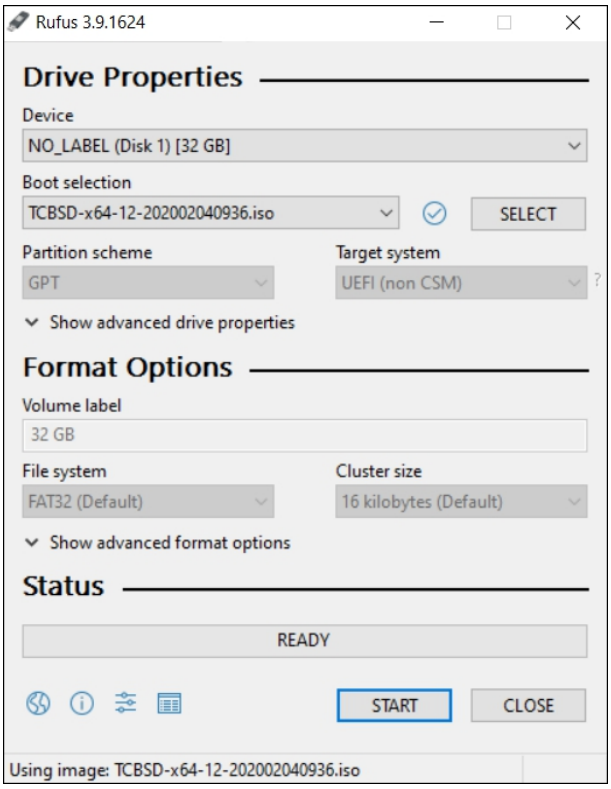
Use a flash tool such as Rufus to do this. You can then start Industrial PC from the USB stick and install TwinCAT/BSD.

Requirements for this step:

- Download Rufus tool from https://rufus.ie/
  Note **:** Note that more recent versions of Rufus are incompatible with Windows 10 version 21H1. It is therefore recommended to use Rufus 3.13.
- USB stick with at least 2 GB capacity.

Proceed as follows:

1. Start the Rufus tool on a PC with Windows operating system.
2. Click **Select** and select the image you want to upload to the USB stick.



3. Under **Device**, select a USB stick as the target drive. If only one external drive is connected to your PC, the USB stick will be selected automatically.
   Note **:** Data on the USB stick is irrevocably deleted.

4. Click **Start** to upload the image to the USB stick.

⇨ The process may take a few minutes. Do not cancel the process until the message **Ready** appears. You have successfully created a bootable USB stick and can install TwinCAT/BSD on the Industrial PC in the next step.

2.2 Check BIOS settings

Check the BIOS settings to be able to start the Industrial PC from the bootable USB stick you created. For TwinCAT/BSD the boot mode in the BIOS must be set to UEFI or Dual Boot. Use Dual Boot if you wish to switch between storage media with different operating systems.

Start the BIOS setup and adjust the boot mode if the settings on your Industrial PC differ.

Proceed as follows:

1. Restart your Industrial PC and press **[Del]** to start the BIOS setup.
   The BIOS setup window appears.
2. Under **Boot > Boot mode select** set the option **UEFI** or **DUAL**.
3. Press **[F4]** to save the settings and exit the BIOS setup.
   The device is restarted.

⇨ You have successfully configured the BIOS and can install the TwinCAT/BSD in the next step.

2.3 Installing TwinCAT/BSD

Connect the bootable USB stick with TwinCAT/BSD image to an Industrial PC and start the device.

Requirements:

- Bootable USB stick with TwinCAT/BSD image.
- Min. 4 GB free disk space.

Proceed as follows:

1. Connect the USB stick with the TwinCAT/BSD image to the Industrial PC.
2. Start the Industrial PC and press **[F7]** to enter the boot menu.
3. Select the UEFI entry for the USB stick and confirm with **[Enter]**.
   The Industrial PC boots from the USB stick, and the Beckhoff TwinCAT/BSD installer is executed.
4. Select the option **TC/BSD Install** to install TwinCAT/BSD .
5. Set a password and follow the further installation instructions.

⇨ Restart the Industrial PC. TwinCAT/BSD is loaded.

3 TwinCAT/BSD

TwinCAT/BSD combines the TwinCAT runtime with FreeBSD, an industrially tested and reliable open source operating system. In addition to multi-core support and a small footprint, TwinCAT/BSD with the Beckhoff Package Server offers a simple way to install TwinCAT Functions and FreeBSD applications or to update the complete system.

What is FreeBSD

FreeBSD is a Unix-compatible open source operating system directly originating from Berkeley Software Distribution (BSD). As an open source project, FreeBSD is continually being developed further, improved and optimized by a large group of developers. On account of the generous BSD license, Beckhoff has opted for FreeBSD, which enables the integration of TwinCAT without licensing problems.

FreeBSD is very popular and is used worldwide by renowned companies. A detailed list of users can be found here:
https://www.freebsdfoundation.org/freebsd/#whois

FreeBSD supports both x86 and X64 platforms and makes scalable systems possible with ARM CPUs extending up to powerful Xeon CPUs.

Further information on FreeBSD can be found on the homepage of the FreeBSD Foundation or that of the FreeBSD project:
https://www.freebsdfoundation.org/
https://www.freebsd.org/

TwinCAT

TwinCAT/BSD supports all TwinCAT 3 runtime functions. The programming is still carried out with the familiar Microsoft Visual Studio®-based TwinCAT XAE from a Windows development computer. TwinCAT/BSD offers multi-core support, allowing individual cores to also be reserved for the exclusive use of TwinCAT.

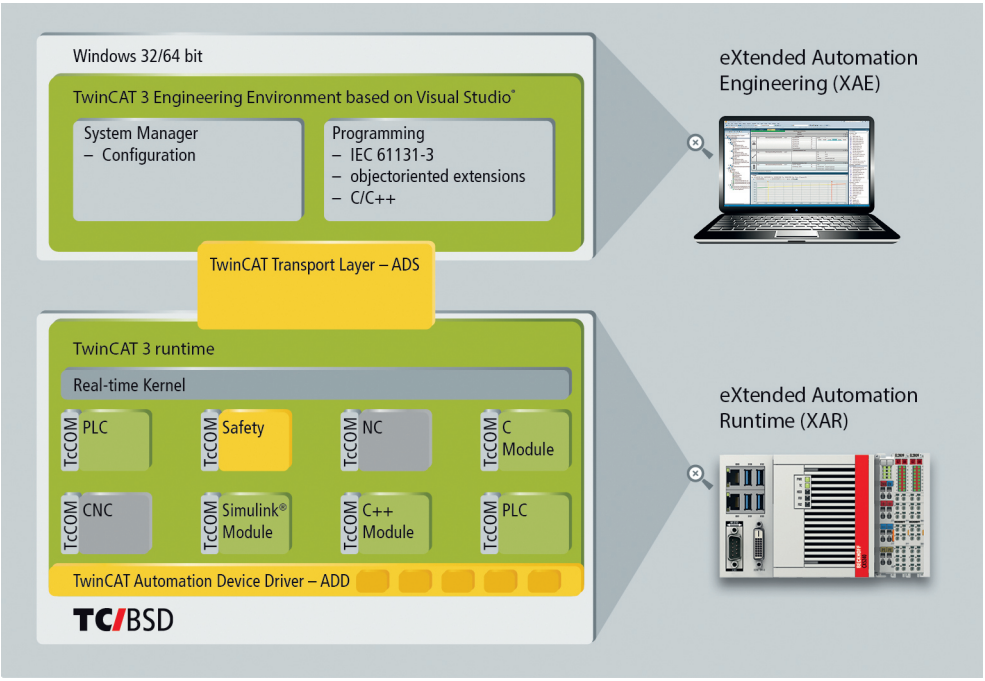


Fig. 1: Structure of the TwinCAT 3 Runtime under TwinCAT/BSD.

In addition to the TwinCAT HMI Server, an HTML5 web browser can be used as a client for TwinCAT HMI. The configuration takes place as usual via the graphic editor of the TwinCAT XAE development environment.

Software and updates

In addition to a large number of FreeBSD programs, TwinCAT Functions can also be installed via the Beckhoff Package Server. Moreover, the uncomplicated updating of the operating system as well as the TwinCAT runtime is possible in this way via the network. Software packages can also be installed offline. The software packages are first loaded to a development computer with a network connection and later installed directly on the Beckhoff Industrial PC. The hosting of the customer's own package server on their side is also possible. Apart from FreeBSD programs that can be offered in this way, many well-known programs from Linux are also available:
https://www.freebsd.org/ports/

Write filter

As is familiar from the Windows operating system, TwinCAT/BSD provides a write filter that protects the system from persistent changes. With the write filter activated, the system is in a previously defined state following a restart.

Backup and restore

A TwinCAT/BSD system can be backed up and restored using a USB stick that offers similar functions as the Beckhoff Service Stick for Windows operating systems. A backup can also be created from the live system, which is backed up locally or via the network to a remote system.

3.1 Credentials

Changing the standard password

For security reasons, change the standard password after logging in for the first time.

When TwinCAT/BSD is delivered, a user (`Administrator`) is available by default, with which you can log on to the console. This user does not have conventional administrator rights like under Windows systems but has the authority to obtain root rights for certain purposes. Use the `doas` command to obtain root rights. `doas` corresponds to the command `sudo`, a command known from other Unix-like operating systems.

Login data:

- Login: `Administrator`
- Password: `1`

Proceed as follows:

1. Start the Industrial PC.
2. Log in with the user name `Administrator` and the password `1`.
3. After successful login, the user and the host name of the Industrial PC is displayed. For example: CX-1D7BD4.
   ```
   Administrator@CX-1D7BD4$
   ```
4. Enter the command `passwd` in order to set a new password for TwinCAT/BSD. Follow the instructions.

⇨ You have successfully logged in and set a new password for TwinCAT/BSD.

3.2 ZFS properties

ZFS is a file system that combines the roles of volume manager and file system. What is special here is that ZFS knows the structure of the storage media and a contiguous memory pool (zpool) is thus available. The memory pool is divided between the available file systems. As soon as more storage media are added to the pool, the existing file systems automatically grow with them and the new storage space is made available to all file systems.

Conventional file systems such as NTFS, ext3 or UFS behave differently. This separates hard disks, RAID controllers, volume managers, and file systems from one another. File systems can only be created on one hard disk at the same time. As soon as a second hard disk is added, two separate file systems have to be created.

Other advantages of ZFS are:

- RAID functionality is available by default.
- Switch-off-proof thanks to copy-on-write
- Automatic data error detection through checksums
- Convenient backup options through snapshots

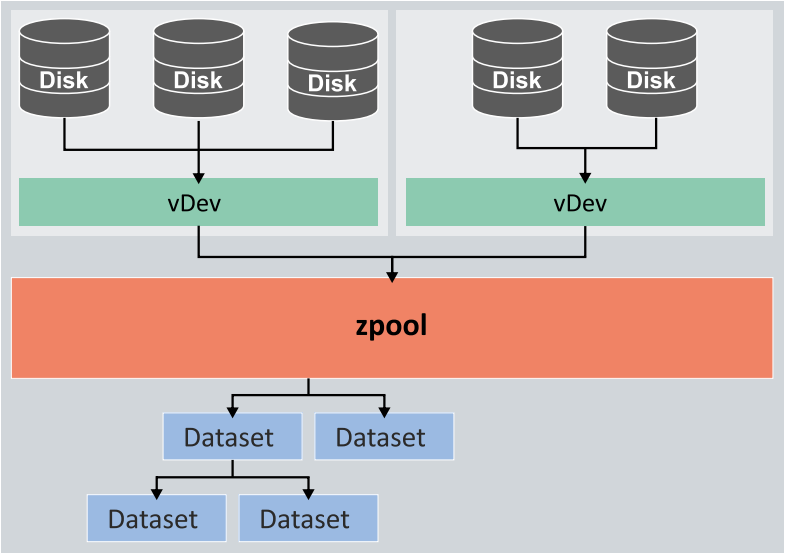


Fig. 2: Overview of the structure of the memory pool, including storage media and datasets.

vDev
The vDevs represent the basic hardware, such as HDDs, SSDs or CFast cards. There are various types of vDev. A vDev can consist of a hard disk, a group of hard disks, a file, a mirror of two or more hard disks or various RAID-Z configurations.

If several vDevs are used, then the data will be divided among the available vDevs in order to increase the speed and to utilize the storage space to the optimum. If one vdev fails, the data of the entire pool is lost. Suitable redundancy (e.g. RAID1) is therefore useful for a vdev.

Memory pool (zpool)
A memory pool (zpool) consists in turn of one or more vDevs. ZFS is based on a memory pool (zpool), which is essentially a collection of vDevs. The vDevs for their part represent the basic hardware, such as HDDs, SSDs or CFast cards, which store the data.

The vDevs are combined into a memory pool. A memory pool is used when one or more file systems (datasets) or block devices (volumes) are to be created. These datasets and volumes share the storage space available in the pool.

Datasets
Dataset is the general term for a ZFS file system, volume, snapshot or clone. Any number of datasets can be created, which are based on a memory pool and contain directories and files. Datasets are hierarchically based on one another. There is a root dataset with following parent datasets, child datasets and further graduations.

The datasets inherit all properties from the parents and grandparents. However, it is also possible to change and overwrite the default values inherited from the parents and grandparents. For each dataset properties such as compression, write and read access, storage space quotas or network shares can be defined.

Example of a dataset:
zroot/tmp

In this example, *zroot/* is the root dataset and also the name of the memory pool (zpool) under TwinCAT/BSD. The command `zfs list` can be used to display all available datasets.

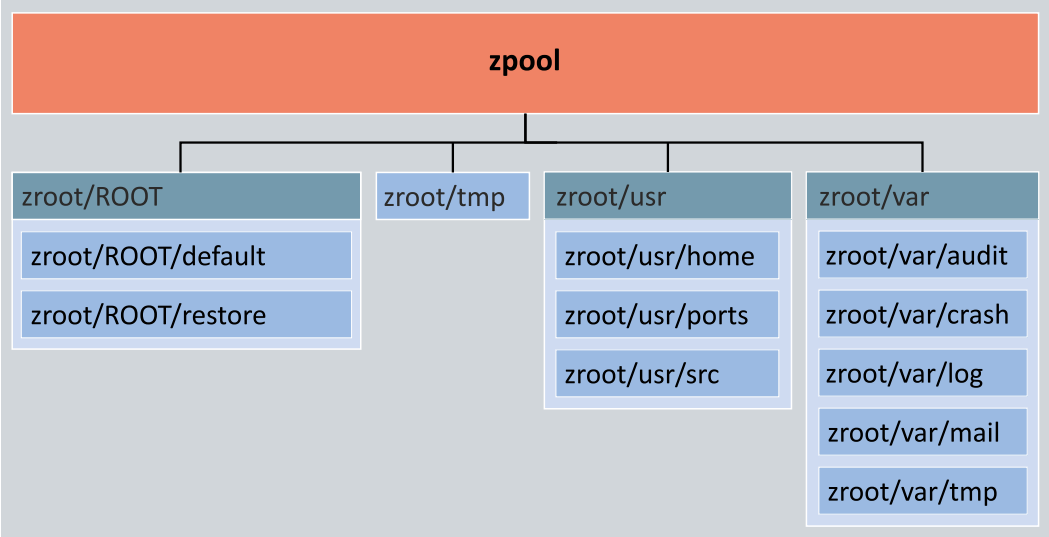


Fig. 3: Datasets of the TwinCAT/BSD operating system.

The dataset `zroot/ROOT/defaul` contains the basic system, all programs and TwinCAT. The dataset `zroot/ROOT/restore` is a boot environment that can be used for restoring recovery points and resetting to factory settings (see Restore options [▶ 76]). The other datasets are mounted at their respective mount points and can be accessed through the file system hierarchy (see Directory structure [▶ 15]). Datasets facilitate customization of options such as read and write permissions for entire memory areas or limitation of storage space for log files or the home directory, for example. Furthermore, individual datasets can be backed up using snapshots.

In addition, `zfs list` specifies the default mount point for each dataset, i.e. the point in the operating system's file system hierarchy through which the dataset can be accessed when it is mounted. Most datasets are mounted automatically directly after system startup. The command `zfs mount` displays the datasets that are currently mounted. A file system, directory or device is only made accessible to the user when a dataset is mounted. The memory pool (zpool) and the associated datasets are mounted in TwinCAT/BSD directly after booting.

Volumes
A volume is a special type of dataset. It is not inserted as a file system and is instead a block device under `/dev/zvol/poolname/dataset`. This allows the volume to be used for other file systems, to provide the hard disks to a virtual machine or to be exported via protocols such as iSCSI or HAST (Highly Available Storage). A volume can be formatted with any file system or it can function as a pure data memory. A volume appears to the user to be a normal disk with a fixed size.

3.3 Directory structure

The directory structure of TwinCAT/BSD is based on the Filesystem Hierarchy Standard (FHS).

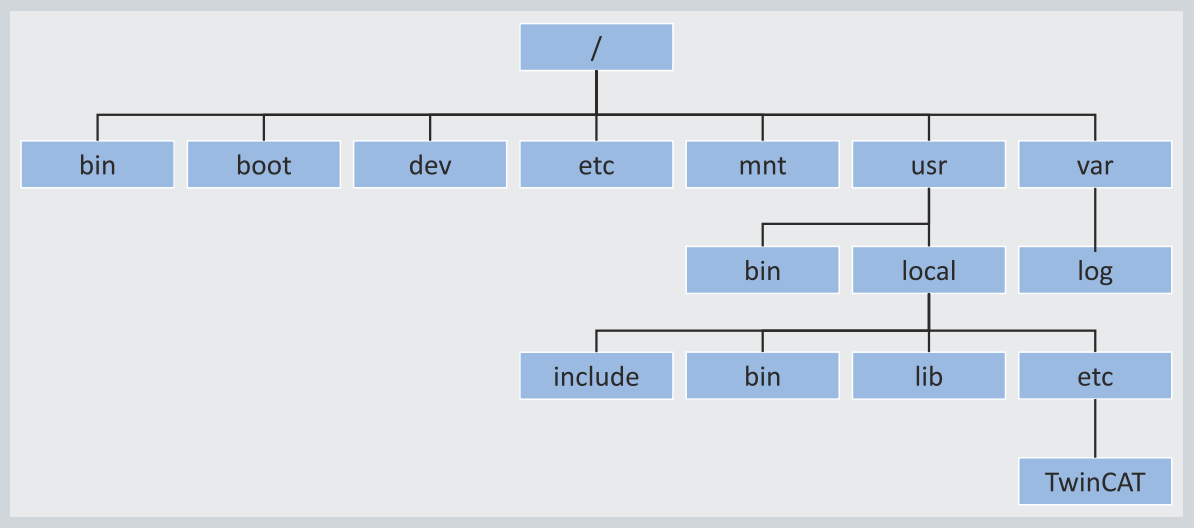


Fig. 4: TwinCAT/BSD directory structure.

The command `cd` can be used to navigate to a different directory. `cd ..` moves up one level in the directory hierarchy, `cd /bin` moves to the directory `/bin`. The command `ls` lists the files in the current directory.

Table 1: Overview of important TwinCAT/BSD directories.

Directory	Description
/	Root directory and top-level directory hierarchy.
/bin/	Basic user applications for single-user and multi-user environments.
/boot/	Kernel, drivers, programs and configuration files for the boot process.
/dev/	Device nodes, which can be used to access hardware directly, for example.
/etc/	System-relevant scripts and configuration files.
/home/	Link to `/usr/home`, where the home directories of the users are located.
/mnt/	Empty directory; usually serves as a mount point for USB sticks, for example.
/root/	Home directory of the superuser `root`.
/sbin/	Basic system applications for single-user and multi-user environments.
/usr/	Unix system resources, contains most of the user applications.
/usr/bin	General applications.
/usr/include/	Contains header files for C compilers.
/usr/local/	Local programs and libraries, i.e., software installed by a user, such as software unrelated to the basic FreeBSD system itself.
/usr/local/bin/	Mainly **Beckhoff applications**
/usr/local/etc/	Configuration files, **TwinCAT directory** with **TwinCAT Functions** and **PLC project**.
/usr/local/include/	Including ADS header files **TcAdsDef.h** and **TcAdsAPI.h**
/usr/sbin/	System applications that are executed by the user.
/var/	Variable files, i.e. temporary files with changing content such as log files.
/var/log/	Contains system log files.

Programs that are located in one of the bin or sbin directories can usually be called from the command line without specifying the path. They are defined in the shells as environment variables.

3.4 Write filter

TwinCAT/BSD has a write filter that protects certain data sets against write access. The advantage of a write filter is that the user can secure a system in a preconfigured state. Following a restart, the system is automatically reset to the originally defined state.

The dataset `zroot/ROOT/default`, which contains most of the system and TwinCAT, is protected against write accesses when the write filter is active. No other datasets are covered by the write filter. For example, user files can still be persistently stored at `/usr/home` or log files at `/var/log`, even if the rest of the system is reset after a restart.

3.4.1 Enabling or disabling the write filter

This step shows how to enable or disable a write filter under TwinCAT/BSD. Note that the changes to the write filter only take effect after a restart.

Proceed as follows:

1. Enter the command `doas service bwf enable` in the console to enable the write filter.
2. Confirm the command with the administrator password.

```
Administrator@CX-3D6912:~ $ doas service bwf enable
Password:
bwf_enable: NO -> YES
writefilter enabled, please reboot to make your changes take effect.
```

3. Restart the Industrial PC with `shutdown -r now` to apply the settings.

⇨ The write filter is active after the restart. The write filter is deactivated again with the command `doas service bwf disable`.

3.4.2 Defining exceptions

Exceptions for the write filter can be defined by creating new datasets, since only the dataset `zroot/ROOT/default` is protected from write accesses; all other system datasets, including newly created datasets, are excluded from the protection.

This chapter shows an example of how a separate dataset can be created for the TwinCAT boot directory, thereby excluding this directory from the write filter protection.

Requirements:

- Save the TwinCAT boot directory in advance if you follow this example.
- Disable the write filter (see Enabling or disabling the write filter [▶ 16]).

Proceed as follows:

1. Enter the command `doas rm -rf /usr/local/etc/TwinCAT/3.1/Boot/*`.
2. The directory `usr/local/etc/TwinCAT/3.1/Boot` is detached from the file hierarchy.
3. Enter the command `doas zfs create -o mountpoint=/usr/local/etc/TwinCAT/3.1/Boot zroot/usr/TwinCAT-Boot` to mount the new dataset `zroot/usr/TwinCAT-Boot`.

⇨ You have successfully created a new dataset for the TwinCAT boot directory. Use `zfs mount` to display all mounted datasets, including the new dataset `zroot/usr/TwinCAT-Boot`. From now on, all directories below this directory are no longer protected from write access by an active write filter.

3.5 Text Editors

The use of text editors is the simplest method of configuring TwinCAT/BSD without additional programs. Text files can be opened and edited in the console using a text editor.

Easy Editor (ee)

With TwinCAT/BSD the Easy Editor (ee) can be used for this task. Enter the command `ee` in the console to start the Easy Editor. Use `ee filename` to open the file to be edited with the name `filename`.

System files are protected for security reasons. Under TwinCAT/BSD they can only be opened by users with extended rights (root rights). Use (doas) `doas ee filename` to open system files with root privileges.

After opening the editor, the most important functions are listed at the top in the display.

```
^[ (escape) menu  ^y search prompt  ^k delete line   ^p  prev li   ^g prev page
^o ascii code     ^x search         ^l undelete line ^n next li    ^v next page
^u end of file    ^a begin of line  ^w delete word   ^b back 1 char
^t top of text    ^e end of line    ^r restore word  ^f forward 1 char
^c command        ^d delete char    ^j undelete char ^z next word
=====line 1 col 0 lines from top 1 ============================================
```

The (^) character represents the **[Ctrl]** key. Therefore, if you wish to use the function ^c you must press the key combination **[Ctrl]** + **[c]**.

Further information about the Easy Editor and its functions can be found in:
https://www.freebsd.org/cgi/man.cgi?query=ee&sektion=1&manpath=freebsd-release-ports

vi editor

TwinCAT/BSD also has more powerful text editors such as the vi editor, which can be used by experienced users. This text editor offers more functionality than the Easy Editor (ee), but is less intuitive.

For more information and functions about the vi editor, see:
https://www.freebsd.org/cgi/man.cgi?query=vi&sektion=1

4 Network settings

4.1 Setting the IP address

DHCP is enabled by default in delivery state. If there is no DHCP server in the network, TwinCAT/BSD automatically assigns an IP address (169.254.x.x)) after a timeout of five seconds. The alternative is a fixed IP address. This step shows how to set a fixed IP address in the console.

These settings are alternatively possible via the web interface of the Beckhoff Device Manager (see: Beckhoff Device Manager: web interface [▶ 42]).

Proceed as follows:

1. Enter `ifconfig` in the console in order to query the network configuration. The Ethernet interfaces `igb0` and `igb1` of an Industrial PC with two interfaces are listed in this sample. The interface `igb1` is active and connected to a network.

```
Administrator@CX-1D7BD4$ ifconfig
igb0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      options=e505bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,TSO4,LRO,
      VLAN_HWFILTER,VLAN_HWTSO,RXCSUM_IPV6,TXCSM_IPV6>
      ether 00:01:05:1d:7b:d4
      media: Ethernet autoselect
      status: no carrier
      nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
igb1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      options=e505bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,TSO4,LRO,
      VLAN_HWFILTER,VLAN_HWTSO,RXCSUM_IPV6,TXCSM_IPV6>
      ether 00:01:05:1d:7b:d5
      inet 172.17.40.22 netmask 0xffffff00 broadcast 172.17.40.255
      media: Ethernet autoselect (100baseTX <full-duplex>)
      status: active
      nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
```

2. Enter `doas ee /etc/rc.conf` in the console.
   The file `rc.conf` opens in the editor.
3. Use the arrow keys to navigate below the entry `ifconfig_default="DHCP"` and add the following line:

```
ifconfig_igb1="inet 172.17.40.30 netmask 255.255.255.0"
```

4. Note the order of the entries in the configuration file. The system reads configuration files from top to bottom. Configuring a static IP address after the DHCP configuration overwrites the previous DHCP configuration. The default in ifconfig_default means that this configuration applies to all interfaces. The following entries can be used to partially or completely overwrite this configuration.
5. Define the IP address with `inet` and the subnet mask for the Ethernet interface `igb1` with `netmask`.
6. Press **[Esc]** and select the option `a) leave editor` and subsequently `a) save changes`.

⇨ You have successfully set `172.17.40.30` as the fixed IP address. Enter the command `doas service netif restart && doas service routing restart` in the console to have the settings applied. Then check the network settings with the command `ifconfig`.

4.2 Changing the host name

This step shows how to change the host name of the Industrial PC. Note that in doing so you will also change the unique name of the Industrial PC in a network. Do not use the prefixes CX- or CP- for your naming convention, otherwise the prefix CX- or CP- and the last 3 bytes of the MAC address will automatically be used as host name.

These settings are alternatively possible via the web interface of the Beckhoff Device Manager (see: Beckhoff Device Manager: web interface [▶ 42]).

Proceed as follows:

1. Enter the command `doas ee /etc/rc.conf` in the console.
   The file *rc.conf* opens.

```
zfs_enable="YES"
# network services and TwinCAT settings
pf_enable="YES"
sshd_enable="YES"
TcSystemService_enable=YES
# custom settings
hostname="CX-1D7BD4"
ifconfig_igb0="DHCP"
ifconfig_igb1="DHCP"
allscreens_kbdflags="-b quiet.off"
# Debugging settings
syslogd_flags="-ss"
#keymap="de.noacc.kbd"
```

2. Change the host name under the entry `hostname="CX-112233"`.
   Press **[Esc]** and save the changes.

⇨ You have successfully changed the host name. The new host name will only be adopted after a restart.

4.3 Firewall

TwinCAT/BSD provides a complete and fully-featured firewall within the package filter (PF). The firewall is factory-set to be restrictive and allows only a few incoming and outgoing connections. The rules for the firewall are stored in a configuration file. You can open the configuration file using the command `doas ee /etc/pf.conf`.

The rules for ports used by Beckhoff services are included through "anchor bhf" in the file `pf.conf` and are created dynamically for TwinCAT Functions. Custom rules for the firewall should still be added to the `pf.conf` file.

Note that the unencrypted ADS port 48898 is disabled by default. Use Secure ADS instead or enable ADS port 48898 with the following entry in the firewall:

Table 2: Firewall rule for unencrypted ADS communication.

Rule	Description
`pass in quick proto tcp to port 48898 synproxy state`	TCP connections on ADS port 48898 (ADS/TCP), disabled by default.

4.3.1 Enable and disable firewall

The firewall is enabled by default. Disabling the firewall can be useful or even necessary in many cases, e.g. in a test environment. This step shows how to disable the firewall. Note that without a firewall, incoming and outgoing connections will no longer be scanned. Never disable the firewall permanently.

Proceed as follows:

1. Enter the command `doas service pf stop` in the console.
   The firewall is disabled.
2. Enter the command `doas service pf start` in the console to re-enable the firewall.

⇨ The firewall is automatically re-enabled after each restart. This behavior is ensured by the entry `pf_enable="YES"` in the `rc.conf` file.

4.3.2 Enable port

Automatic port enabling for TwinCAT Functions

Ports that are required for TwinCAT Functions are automatically enabled once the TwinCAT Functions have been installed.

This step shows how to enable a TCP port. As an example, an incoming connection is created for TCP port 502, which is required for Modbus/TCP communication.

Enable a port as follows:

1. Enter the command `doas ee /etc/pf.conf` on the console.
   The `pf.conf` configuration file is opened.
2. Create the `pass in quick proto tcp to port 502 keep state` rule to enable TCP port 502.
3. Press **[Esc]** and save the changes.
4. Enter the command `doas pfctl -f /etc/pf.conf` to reload the rules. The firewall must be activated for this.

⇨ You have successfully enabled a port. Use the command `doas pfctl -f /etc/pf.conf` to enable the rules immediately. Otherwise, the rule takes effect after the next firewall restart.

4.4 WLAN

4.4.1 Connecting to WLAN

This step shows how to establish a WLAN connection with an access point under TwinCAT/BSD. In addition, you will learn how to search for WLAN networks and determine the SSID.

WLAN is encrypted with WPA2, and the IP address is automatically assigned by a DHCP server.

Requirements:

- Beckhoff WLAN sticks: CU8210-D001-0101 or CU8210-D001-0102
- SSID and password of an existing WLAN.

Proceed as follows:

1. Enter the command `sysctl net.wlan.devices` in the console to obtain the device name. With a Beckhoff WLAN stick, for example, `rtwn0` is output as the device name.
2. Open the file `rc.conf` with `doas ee /etc/rc.conf` and add the following lines:

```
# wireless
wlans_rtwn0="wlan0"    #wlan0 is now your network interface
ifconfig_wlan0="WPA DHCP country DE"
```

3. If DHCP is not active or desired, a fixed IP address is defined with the following entry.

```
# wireless
wlans_rtwn0="wlan0"    #wlan0 is now your network interface
ifconfig_wlan0="WPA inet 192.168.0.100 netmask 255.255.255.0 country DE"
```

4. Restart the network service with `doas service netif restart` to apply the settings in the rc.conf file.
5. Search for new WLAN networks using `doas ifconfig wlan0 up scan`. The command `doas ifconfig wlan0 list scan` displays networks that are already known.
6. Save the access data for a WLAN network with the command `doas ee /etc/wpa_supplicant.conf` by adding the following lines to the file `wpa_supplicant.conf`.

```
network={
    ssid="myssid"    #for myssid specify the name of the network
    psk="mypsk"      #for mypsk enter password of network
}
```

7. Enter `doas service netif restart` to restart the network service.

⇨ The WLAN connection is established. Use `ifconfig` to display the network status of the WLAN interface.
For further information, see: https://www.freebsd.org/doc/handbook/network-wireless.html

4.4.2 Configuring as access point

You can configure an Industrial PC as an access point under TwinCAT/BSD. This feature requires the `hostapd` packet to be installed. Install the `hostapd` packet with: `doas pkg install hostapd`

The `hostapd` daemon takes care of client authentication and key management on the WPA2-enabled access point.

Make sure the correct regulatory domain is used for the respective country. This includes, for example, the permitted channels, permitted transmission power and DFS activation for certain 5 GHz channels.

Requirements:

- Installing the `hostapd` packet.
- Internet connection.
- Reinsert the WLAN stick if it was plugged in at startup.

Proceed as follows:

1. Open the file `rc.conf` with `doas ee /etc/rc.conf` and add the following lines:
   Example:

```
hostapd_enable="YES"    #starts the hostapd daemon automaticaly after boot
wlans_rtwn0="wlan0"    #wireless interface used for access point
create_args_wlan0="wlanmode hostap ssid yourSSIDname authmode WPA2"
ifconfig_wlan0="inet 192.168.0.1 netmask 255.255.255.0 country DE"
```

2. Use the command `doas ee /etc/hostapd.conf` to open the file `hostapd.conf` and add the following lines:
   Example:

```
interface=wlan0
debug=1
ctrl_interface=/var/run/hostapd
ctrl_interface_group=wheel
ssid=yourSSIDname
wpa=2
wpa_passphrase=freebsdmall    #password for wlan network
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
channel=6    #Channel for the desired radio band (default: 0 stands for ACS, automatic Channel
Selection)
hw_mode=g    #Operation mode, in this case g=IEEE802.11g (2.4 GHz)
country_code=DE #used to set the right regulatory domain for your country
ieee80211d=1    #advertises the country_code an the set of allowed channels and transmit power
levels based on the regulatory limits (default=0)
```

3. Enter the command `doas service hostapd forcestart` to start the access point.

⇨ You have successfully configured the Industrial PC as an access point. WLAN devices can now connect to the network using the SSID and the password from the `hostapd.conf` file.
For further information see: https://www.freebsd.org/doc/handbook/network-wireless.html

4.4.3 Setting up a DHCP server

Depending on the network infrastructure, you may need a DHCP server. This step shows how to install and configure a DHCP server.

Requirements:

- Internet connection.
- Adjust the firewall and apply the rules to the WLAN interface (see: Firewall [▶ 19])

Proceed as follows:

1. Enter the command `doas pkg install dhcpd` in the console to install the DHCP server.
2. With `doas ee /usr/local/etc/dhcpd.conf`, open the file `dhcpd.conf`. and edit the configuration to suit your requirements.
   Example:
   ```
   subnet 192.168.0.0 netmask 255.255.255.0 {
   range 192.168.0.10 192.168.0.20;
   default-lease-time 600;
   max-lease-time 72400;
   option subnet-mask 255.255.255.0;
   }
   ```
3. Open the file `rc.conf`. with `doas ee /etc/rc.conf` and add the following lines:
   ```
   dhcpd_enable=”YES”
   dhcpd_flags=”wlan0”
   dhcpd_ifaces=”wlan0”
   ```
4. Enter the command `doas service dhcpd start` to start the DHCP service. After a restart, the DHCP service is started automatically because of the corresponding entry in `rc.conf`.
5. The command `ee /var/db/dhcpd.leases` can be used to view current and expired leases.

⇨ The DHCP server is then active and only listens to requests on the `wlan0` interface.
For further information see: https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-dhcp.html or https://man.openbsd.org/dhcpd.8

5 System update

TwinCAT/BSD release process

The system update is performed by default via the preset Beckhoff Package Server or the Beckhoff repository, which provides all required packages (see: Package Server [▶ 27]).

As a rule, a new version of the Beckhoff repository is published on the first Tuesday of each month. Provided that all internal tests are passed successfully. If the internal tests fail, the release will be postponed to the next month.

Only the Beckhoff repository is updated, which does not necessarily mean that a new TwinCAT/BSD or TwinCAT version is automatically published in each new repository version. It could also simply be individual packages, third-party packages, or packages that you don't even have installed on your system. When performing an update, it is recommended to update the entire system and not just individual packages. This is because the packages in the Beckhoff repository are tested as a complete system in each release, thus avoiding incompatibilities.

It is possible to download and save a tested version of the Beckhoff repository as a local repository at any time. This allows you to freeze your system in a tested state and use it for series machine construction, for example. In this case, the local repository and all the packages it contains can be made available via a server in the local network or via a USB stick (see: Set up local repository [▶ 31]).

Call TwinCAT/BSD version

The TwinCAT/BSD version can be called with the command `TcSysExe.exe`. There are different version information that you can get from the system. The current version is listed under the TC/BSD entry:

```
Administrator@CX-0C8432:~ $ TcSysExe.exe

The software licenses can be found in this folder: /usr/local/etc/TwinCAT/3.1/System/Legal/
TcOsSys.dll: TcOsSys_Rel31_4024_20210804.2
TwinCAT Build: 3.1.4024.19
AMS Net Id: 5.12.132.50.1.1
TC/BSD: 13.0.8.2,2
Administrator@CX-0C8432:~ $
```

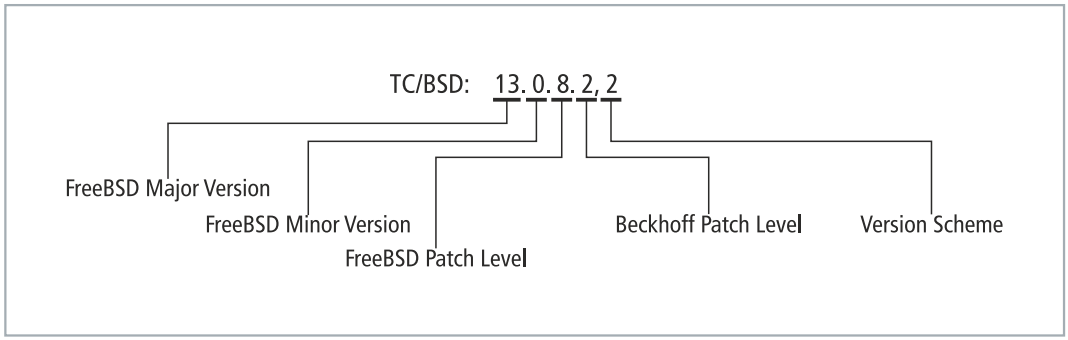


Fig. 5: Breakdown of the TwinCAT/BSD version.

Call Beckhoff repository version

The command `pkg info os-generic-userland-conf | grep Version` displays the pipeline number for the corresponding internal Beckhoff repository. In this sample the pipeline ID 34729 is displayed after the TwinCAT/BSD version:

```
Administrator@CX-3AE2C6:~ $ pkg info os-release-bhf | grep Version
Version : 13.0.11.3_55702
Administrator@CX-3AE2C6:~ $
```

Perform update

There are two ways to update the system. The first option is to update TwinCAT/BSD to the latest version. In the second, the entire base system is updated and thus the latest major version is installed.

- Update TwinCAT/BSD [▶ 26]
- Update major version [▶ 26]

Update TwinCAT/BSD

The command `pkg upgrade` is used to update packages. In doing so, `pkg upgrade` compares the versions of all installed packages with the versions available in the configured package repositories. The Beckhoff Package Server mirrors most of the packages from the normal FreeBSD repository and also contains the following specific Beckhoff packages that can be updated:

- TwinCAT/BSD updates
- TwinCAT 3 updates
- TwinCAT 3 functions

In addition, `pkg upgrade` also tries to update the dependencies of the packages. However, no new packages will be installed except when necessary to satisfy package dependencies.

Individual packages can also be updated, but it is recommended to update the entire system to avoid incompatibilities. This is because the packages in the Beckhoff repository are tested as a complete system in each release.

Update major version

The latest major version is required because after a change to the next higher major version, all security updates are provided only for the current version. Older versions are no longer maintained.

In addition to this, new features may be implemented and made available in the future in the latest major version, which will be required for TwinCAT/BSD.

5.1 Update TwinCAT/BSD

Creating a restore point

Create a restore point before making a major system change or updating programs (see: Restore options [▶ 76]).

This step illustrates how to update TwinCAT/BSD. The goal is to bring the system up to date. Create a backup of your system before you update the system.

Update the system as follows:

1. Enter the command `doas pkg upgrade` in the console.
   The packages to be updated are displayed.

```
Checking for upgrades (27 candidates): 100%
Processing candidates (27 candidates): 100%
The following 1 package(s) will be affected (of 0 checked):

Installed packages to be UPGRADED:
      ca_root_nss: 3.47_1 -> 3.47.1

Number of packages to be upgraded: 1

287 KiB to be downloaded.
```

2. Enter the command `doas pkg upgrade` upgrade on the console, in order to update all packages or `doas pkg upgrade <packagename>` in order to update a specific package.

⇨ Take a close look at the packages to be updated. If there are any ambiguities, read the release information for the package and only then start the update.
Define exceptions and lock packages that should not be updated (see: Lock [▶ 31]). Locked packages are not updated, uninstalled or reinstalled.

5.2 Update major version

This chapter describes how to update a major version from TwinCAT/BSD 12 to TwinCAT/BSD 13, for example. The latest major version is required for security updates, among other things.

The update is carried out with the aid of a script, because the entire basic system as well as the kernel adapted by Beckhoff is updated. The script also automatically creates a restore point that you can jump back to (see: Resetting to the restore point [▶ 78]).

Proceed as follows:

1. Update the latest major version with `doas pkg update -f && doas pkg upgrade`.
2. Install the script for the update with the command `doas pkg install tcbsd-upgrade`
3. Execute the script with the command `doas tcbsd-upgrade major`. At the end of the process, a message about a successful update to the next major version is displayed.

```
Successfully upgraded to TC/BSD 13 and activated the new BE "upgrade-2022-02-17T15:02:01Z"
Reboot is required.
```

4. Restart TwinCAT/BSD with `doas shutdown -r now` after a successful update.

⇨ After the restart, the new major version is available and displayed after the login.

```
Last login: Thu Feb 17 15:07:47 on pts/0
FreeBSD 13.0-RELEASE-p7 n244930-e4e6bcfbb68- BHF

The software licenses can be found in this folder: /usr/local/etc/TwinCAT/3.1/System/Legal/
TcOsSys.dll: TcOsSys_Rel31_4024_20211129.2
TwinCAT Build: 3.1.4024.22
AMS Net Id: 5.61.105.18.1.1
TC/BSD: 13.0.7.2,2
```

6 Package Server

The Beckhoff Package Server is a server hosted by Beckhoff that contains a collection of precompiled software – the so-called packages. This is the easiest way to install additional software under TwinCAT/BSD or update existing software.

During installation TwinCAT/BSD accesses the preset Beckhoff repository:
https://tcbsd.beckhoff.com/TCBSD/13/stable/packages/All/

The Beckhoff package server mirrors a large part of the normal FreeBSD repository and also includes the following specific Beckhoff packages:

- TwinCAT/BSD updates
- TwinCAT 3 updates
- TwinCAT 3 functions

Supported packages
Unlike Beckhoff packages, third-party packages are not checked and are not supported.

Repositories

You can fall back on additional repositories if you want to use other packages that are not provided by Beckhoff. In addition to the standard Beckhoff repository, you can add, activate, deactivate or remove additional repositories.

Make sure you configure additional repositories in a file of their own under `/etc/pkg`. TwinCAT/BSD has two repositories on delivery:

- `TCBSD.conf` is the standard Beckhoff repository

```
TCBSD: {
    url: "https://tcbsd.beckhoff.com/TCBSD/13/stable/packages"
    enabled: true,
    signature_type: "fingerprints",
    fingerprints: "/usr/share/keys/bhf-pkg"
}
```

- `FreeBSD.conf` is the official FreeBSD repository and is deactivated by default

```
FreeBSD: {
    url: "pkg+http://pkg.FreeBSD.org/${ABI}/quarterly",
    mirror_type: "srv",
    signature_type: "fingerprints",
    fingerprints: "/usr/share/keys/pkg",
    enabled: yes
}
```

6.1 Switching to a Chinese server

The Beckhoff repository is not accessible from China and has to be switched over to the Chinese server. Run the `pkgrepo-set` script to switch from the default server to the Chinese server.

Switch to the Chinese server as follows:

1. Enter the command `doas sh /usr/local/share/examples/bhf/pkgrepo-set.sh china` in the console.
2. The URL in the file `TCBSD.conf` is changed from `https://tcbsd.beckhoff.com/TCBSD/13/stable/packages` to `https://tcbsd.beckhoff.com.cn/TCBSD/13/stable/packages`.

⇨ You have successfully set the Chinese server. You can restore the default settings with the command `doas sh /usr/local/share/examples/bhf/pkgrepo-set.sh release`.

6.2 Switching to the FreeBSD repository

This step shows you how to switch from the standard Beckhoff repository to the official FreeBSD repository.

Note that packages not installed from a Beckhoff repository may not be compatible with TwinCAT/BSD and may not function properly. The reason for this is that the FreeBSD basic system has been modified for TwinCAT.

Switch to the FreeBSD repository as follows:

1. Enter the command `doas ee /usr/local/etc/pkg/repos/FreeBSD.conf` in the console.
2. Set the variable `FreeBSD: {enabled: no}` to “yes”.

You have successfully changed the package server. From now on, all packages will be loaded from the official FreeBSD repository. Set the variable to "no" to use the standard Beckhoff repository again.

6.3 Package management

A TwinCAT/BSD installation includes the software required for the operating system and the TwinCAT 3 Runtime. You can install additional software or TwinCAT Functions.

You have to decide which functions you need and choose software that provides these functions. This chapter shows you how to:

- search for,
- install,
- update,
- uninstall
- and lock packages, i.e. precompiled software.

The command `pkg info` lists all packages installed on the system, together with their respective versions. `pkg info <packagename>` displays information about a specific package.

```
---snipped---
Administrator@CX-3B151A:~ % pkg info IPC-Diagnostics
IPC-Diagnostics-3.1.4024.5_2019110615523410164
Name           : IPC-Diagnostics
Version        : 3.1.4024.5_2019110615523410164
Installed on   : Fri Nov 8 10:55:37 2019 UTC
---snipped---
```

6.3.1 Search

Before you install software, you can determine whether the software is available on the Package Server. Note that the search is not case sensitive. A distinction is made between case and lower case only with the suffix `-C`.

Proceed as follows:

1. Enter the command `pkg search <packagename>` in the console. Example:

```
Administrator@CX-3B151A:~ % pkg search docbook
docbook-1.5                    Meta-port for the different versions of the DocBook DTD
docbook-sgml-4.5_1             DocBook SGML DTD
docbook-xml-5.0_3              DocBook XML DTD
docbook-xsl-1.79.1_1,1         XSL DocBook stylesheets
sdocbook-xml-1.1_2,2           "Simplified" DocBook XML DTD
Administrator@CX-3B151A:~ %
```

2. Several results are displayed.
3. You can view additional information about the package with the command `pkg search -R docbook`.

```
---snippet---
name: "docbook"
origin: "textproc/docbook"
version: "1.5"
comment: "Meta-port for the different versions of the DocBook DTD"
maintainer: doceng@FreeBSD.org
www: http://www.oasis-open.org/docbook/
abi: "FreeBSD:12:*"
arch: "freebsd:12:*"
---snippet---
```

4. Use the additional information to choose the right software.

⇨ In the next step you can install the software.

6.3.2 Install

Creating a restore point

Create a restore point before making a major system change or installing programs (see: Restore options [▶ 76]).

This step shows you how to install new software under TwinCAT/BSD. The software must be available on the Package Server (see: Search [▶ 29]). You can install additional software or TwinCAT Functions.

With each installation, the `pkg` program checks whether the local data stock matches that on the Package Server; it is updated if necessary.

Proceed as follows:

1. Enter the command `doas pkg install <packagename>` in the console. Sample:

```
---snippet---
Administrator@CX-3B151A:~ % doas pkg install docbook
Password:
Updating FreeBSD12-pkgbase repository catalogue...
FreeBSD12-pkgbase repository is up to date.
All repositories are up to date.
---snippet---
```

2. The `pkg` program automatically searches for additional packages that are required for the installation.
3. Confirm the installation with **[y]**.

⇨ The package is retrieved from the repository and installed on the system. Some packages contain installation messages that contain instructions, warnings and helpful notes.

6.3.3 Update

Creating a restore point

Create a restore point before making a major system change or updating programs (see: Restore options [▶ 76]).

This step illustrates how to update packages under TwinCAT/BSD. Create a backup of your system and start a trial run with the suffix `-n` before you update packages. All the packages that can be updated without performing the installation are listed in the trial run.

Update packages as follows:

1. Type the command `doas pkg upgrade -n` in the console.
   The packages to be updated are displayed.

```
Checking for upgrades (27 candidates): 100%
Processing candidates (27 candidates): 100%
The following 1 package(s) will be affected (of 0 checked):

Installed packages to be UPGRADED:
      ca_root_nss: 3.47_1 -> 3.47.1

Number of packages to be upgraded: 1

287 KiB to be downloaded.
```

2. Enter the command `doas pkg upgrade` upgrade on the console, in order to update all packages or `doas pkg upgrade <packagename>` in order to update a specific package.

⇨ Take a close look at the packages to be updated. If there are any ambiguities, read the release information for the package and only then start the update.
Define exceptions and lock packages that should not be updated (see: Lock [▶ 31]). Locked packages are not updated, uninstalled or reinstalled.

6.3.4 Uninstall

This step shows you how to uninstall software under TwinCAT/BSD. The `pkg` program ensures that an uninstall has no negative consequences for the system and pays attention to dependencies among the packages.

If you uninstall software on which another software depends, that software will also be removed automatically.

```
docbook: 1.5
sdocbook-xml: 1.1_2,2
xmlcatmgr: 2.2_2
docbook-xml: 5.0_3
xmlcharent: 0.3_2
docbook-sgml: 4.5_1
iso8879: 1986_3
```

The `docbook` package depends on the `iso8879` package. If you uninstall the `iso8879` package, `docbook` will also be removed. You can suppress this behavior with the suffix `-f` and uninstall only the selected package without the system paying attention to the dependencies.

Proceed as follows:

1. Enter the command `doas pkg delete <packagename>` in the console. Example: `doas pkg delete iso8879`
2. TwinCAT/BSD lists packages that can be deleted and considers the dependencies.

```
---snipped---
Administrator@CX-3B151A:~ % doas pkg delete iso8879
Password:
Checking integrity... done (0 conflicting)
Deinstallation has been requested for the following 3 packages (of 0 packages in the universe):

Installed packages to be REMOVED:
    iso8879-1986_3
```

```
    docbook-sgml-4.5_1
    docbook-1.5
---snipped---
```

3. Confirm with **[y]** to uninstall the listed packages.

⇨ In conclusion, the uninstalled packages are summarized. Note that unwanted packages can accumulate over time, for example, if you uninstall software with the suffix `-f` or if new software versions have different dependencies.

The command `pkg autoremove` identifies, lists and proposes the uninstallation of unnecessary packages and dependencies. Read the list carefully. Important packages can also be locked so that they are not inadvertently uninstalled (see: Lock [▶ 31]).

6.3.5 Lock

Packages can be locked to prevent them from being inadvertently uninstalled or updated during updates. Locked packages are not updated, uninstalled or reinstalled.

This feature can be especially useful for certain TwinCAT Functions in order to prevent unwanted updates. Locking does not prevent someone with root rights from manipulating the files contained in the package.

Lock packages as follows:

1. Enter the command `doas pkg lock <packagename>` on the console.

```
docbook-1.5: lock this package? [y/N]: y
Locking docbook-1.5
```

2. Confirm the query with **[y]** to lock the package.

⇨ The package remains locked until you unlock it with the command `doas pkg unlock <packagename>`. You can display all locked packages with the command `pkg lock -l`.

```
Currently locked packages:
docbook-1.5
```

6.4 Set up local repository

To install software and updates offline, it is a good idea to use a local repository. The local repository can then be made available via a server on the local network or via a USB stick.

For this purpose, the Beckhoff repository is downloaded first. This can then either be copied to the FAT partition of a USB stick or made available on the network via an FTP or web server. By mirroring the entire content, the package verification also remains the same. For your TwinCAT/BSD system it is as if the packages continue to be downloaded from the Beckhoff repository.

The following describes how to extend a USB stick with a local TwinCAT/BSD repository:

Requirements:

- Program for recursively downloading files from the Internet. An example is the program `wget` which is available for Linux and TwinCAT/BSD.

Proceed as follows:

1. Download the TwinCAT/BSD repository from the Beckhoff server. If you use Linux or TwinCAT/BSD, use the command `wget --recursive --timestamping --level=inf --no-cache --no-parent --no-cookies --no-host-directories --relative --directory-prefix /tmp/mirror https://tcbsd.beckhoff.com/TCBSD/13/stable/packages/`
2. Insert the USB stick into your TwinCAT/BSD device. Note that the device may boot directly from the USB stick according to your BIOS setting. Then plug in the USB stick only after booting.
3. Connect the USB stick to your system (see: Integrating a USB stick [▶ 38]). Automount was deactivated ex works for security reasons.

4. Copy the repository from the directory `/tmp/mirror` to the FAT partition of a USB stick using the command `cp -r /tmp/mirror /mnt/usb`. The path `/mnt/usb` is the mount point of the USB stick in this example.
5. Next, the repository path that points to this Beckhoff Package Server by default must be changed: https://tcbsd.beckhoff.com/TCBSD/13/stable/packages/
6. In this example, the repository path must point to the USB stick in order to use the local repository on the USB stick. Enter the following command: `doas sh /usr/local/share/examples/bhf/pkgrepo-set.sh file:///mnt/mirror/TCBSD/13/stable/packages`

⇨ Use the pkg tool as usual. The command `doas pkg upgrade` can now be used to update and install packages from the USB stick. Instead of the Beckhoff Package Server, the local repository on the USB stick is used. Note that after a reboot the USB stick must be integrated manually again.

7 Configuration

7.1 System information

TwinCAT/BSD provides different system information. The most important system information required for daily work with TwinCAT/BSD and TwinCAT can be retrieved with the tool `TcSysExe.exe` and the TwinCAT Registry

TcSysExe.exe

With TcSysExe.exe it is for example possible to control the TwinCAT mode from the console and to put TwinCAT into Run or Config mode. Call all available parameters with `TcSysExe.exe -help`. The following system information is particularly important:

- The command `TcSysExe.exe` or TcSysExe.exe -version lists information about the used TwinCAT build, the AMS Net Id and the TwinCAT/BSD version:

```
The software licenses can be found in this folder: /usr/local/etc/TwinCAT/3.1/System/Legal/
TcOsSys.dll: TcOsSys_Rel31_4024_20220407.2
TwinCAT Build: 3.1.4024.29
AMS Net Id: 5.66.247.12.1.1
TC/BSD: 13.0.11.1,2
```

- The command `TcSysExe.exe --osImageVersion` shows the TwinCAT/BSD version:

```
Administrator@CX-42F70C:~ $ TcSysExe.exe -osImageVersion
TC/BSD: 13.0.11.1,2
```

- The command `TcSysExe.exe --platformid` shows the TwinCAT 3 platform level of the Industrial PC used:

```
Administrator@CX-42F70C:~ $ TcSysExe.exe -platformid
HW Platform: 70
```

- The command `TcSysExe.exe --netid` displays the AMS Net Id of the Industrial PC:

```
Administrator@CX-42F70C:~ $ TcSysExe.exe -netid
AMS Net Id: 5.66.247.12.1.1
```

TcRegistry.xml

Extensive system settings are possible via the TwinCAT registry. The file is located in the directory `/usr/local/etc/TwinCAT/3.1/TcRegistry.xml` and can be opened and edited with `doas ee /usr/local/etc/TwinCAT/3.1/TcRegistry.xml`. The following system information is particularly important and can be edited in the XML file:

- AmsNetID: Changing the AMS NetID [▶ 71]
- HeapMemSize: Increase heap memory [▶ 73]
- LockedMemSize: Adapting the router memory [▶ 74]

7.2 User and rights management

There are three account types, each of which is subject to different restrictions and is fundamental for account management under TwinCAT/BSD. The following account types are available under TwinCAT/BSD:

- Superuser account
- User accounts
- System user

The **superuser account**, also called `root`, can operate without restrictions. Unlike normal user accounts, `root` has absolute control over TwinCAT/BSD. To ensure system integrity and security, `root` is disabled by default. This means it is not possible to log in directly as `root`.

User accounts are available for normal users who require access to TwinCAT/BSD. They are assigned a unique user name and a home directory and can customize their own user environment. The user `Administrator` is created by default. This user does not have conventional administrator rights like under Windows systems but has the authority to obtain root rights for certain purposes.

The **system users** can start services and programs such as mail or web servers. This makes it possible to restrict programs or services or to enable access rights for certain tasks.

Root rights

Since it is not possible to log in as `root`, users can be assigned root rights in order to operate without restrictions under TwinCAT/BSD. Use the `doas` command to obtain root rights. `doas` corresponds to the command `sudo`, a command known from other Unix-like operating systems.

Groups

TwinCAT/BSD allows user accounts to be grouped together so that their permissions for using individual functions or software can be managed centrally. Instead of assigning the same individual rights to many user accounts, a user role is defined that contains the rights to be assigned. The groups are identified by the group name and the group ID (gid). The TwinCAT/BSD kernel decides on the basis of the user ID (uid) and the group membership of a process whether or not it gives permission to the process.

The file `group` contains all group information, such as group name, group password, group ID and a list of members of the respective group. Call up the file with `cat /etc/group`:

```
wheel:*:0:root,Administrator
```

This excerpt shows the first line of the file `group`. The file is divided into four fields, separated by colons. The first field contains the group name (wheel), the second field contains an encrypted password (*), the third field contains the group ID (0) and the fourth field contains a list with the associated members (root, administrator).

Each user can determine their group affiliation with `id`. Here is an example for the user `Administrator`, who belongs to the groups `1001 Administrator` and `0 wheel`:

```
uid=1001(Administrator) gid=1001(Administrator) groups=1001(Administrator),0(wheel)
```

Use `doas pw groupadd <groupname>` to create a new group. Use `pw groupshow <groupname>` to display a group. Use the command `doas pw groupmod <groupname> -M <username>` to add a user to a group.

```
Administrator@CX-3B151A:~ % pw groupshow AI
AI:*:1007:Skynet,DeepThought,Ava,HAL
```

7.2.1 Create new user

Use the `adduser` command to create new users under TwinCAT/BSD. You will be guided interactively through the process. Only the superuser (root) can create new users. Therefore, you have to run the `adduser` command as administrator and with root rights.

These settings are alternatively possible via the web interface of the Beckhoff Device Manager (see: Beckhoff Device Manager: web interface [▶ 42]).

Requirements:

- Access to the administrator account

Create new users as follows:

1. Enter the command `doas adduser` in the console.
   Confirm the command with the administrator's password.
2. Specify a user name and enter the full name.

```
Username: NewUser
Full name: John Doe
```

3. In the next step you can set the `Uid`, `Login group`, the secondary `Login group` and the `Login class`. Press **[Enter]** to apply default settings.

```
Uid [Leave empty for defaults]:
Login group [NewUser]:
Login group is NewUser. Invite NewUser into other groups? []:
Login class [default]:
```

4. Under `Shell (sh csh tcsh nologin) [sh]:` select a shell with which the user logs in. By default, the Bourne shell (sh) is used for scripts and the command line. However, for direct interaction via the command line, tcsh is particularly suitable as a shell thanks to its better clarity and a larger range of functions.

```
Shell (sh csh tcsh nologin) [sh]: tcsh
```

5. For the following parameters press **[Enter]** to apply the default settings.

```
Home directory [/home/NewUser]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
```

6. In this step you can decide whether a password for the user should be generated randomly or whether you want to assign a password yourself.

```
Use a random password? (yes/no) [no]:
```

7. Then press **[Enter]** to apply the default settings under `Lock out the account after creation? [no]:`.

⇨ A summary is then issued. You can check your entries and create another user if required. In addition, the generated user password is displayed.

```
adduser: INFO: Successfully added (NewUser) to the user database.
adduser: INFO: Password for (NewUser) is: Luq39oGIwPhjT
Add another user? (yes/no): no
Goodbye!
```

7.2.2 Edit user information

The program `chpass` can be used to edit further user information. A superuser (root) has extended rights and can edit more fields than other users.

A superuser (root) can change the standard personal settings, the home directory, the Uid, Gid and the login name. In addition, security settings can be implemented by using `Change` to set the time for password changes or `Expire` to set an expiration time for the user account.

```
#Changing user information for NewUser.
Login: NewUser
Password: $6$q1X1/ZB/NGu9ulrF$.JYhoCPsGT6hk0GD34oJYWwOKGxY67ka818lpy/0HY.7XvXK69
JdeYOtMkNJNQvqBTTfblYBQ3SZ.MYxChPeQ1
Uid [#]: 1002
Gid [# or name]: 1002
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/NewUser
```

A standard user account can edit the first and last name, the shell and the contact information.

```
#Changing user information for NewUser.
Shell: /bin/tcsh
Full Name: John Doe
Office Location:
Office Phone:
Home Phone:
Other information:
```

Edit user information as follows:

1. Enter the command `chpass NewUser` in the console. Or enter `doas chpass NewUser` if you want to edit the information as superuser (root).
2. Use the arrow keys to navigate to the line you want to edit.
3. Press **[Esc]** to switch the editor to command mode.
4. Delete preset values with **[x]**.
5. Press **[i]** to add new text to the cursor.
6. Press **[Esc]** and type `:wq` in the console to save the changes and exit the editor. Or type `q!` to exit the editor without saving.

⇨ Afterwards, a message is issued to indicate that all changes have been saved successfully. If the entries are incorrect, an error message is displayed with a reference to the location.

```
/etc/pw.6RnflE: 15 lines, 412 characters.
chpass: upper-case letters are dangerous in a login name
chpass: user information updated
Administrator@CX-3B151A:~ %
```

7.2.3 Deleting a user

The program `rmuser` can be used to completely remove user accounts from the system. User accounts can only be deleted by the superuser (root).

These settings are alternatively possible via the web interface of the Beckhoff Device Manager (see: Beckhoff Device Manager: web interface [▶ 42]).

Requirements:

- Access to the administrator account

Proceed as follows:

1. Enter the command `doas rmuser NewUser` in the console.
   Confirm the command with the administrator's password.
2. The user account is displayed. Check the data before continuing.

```
Administrator@CX-3B151A:~ % doas rmuser NewUser
Password:
Matching password entry:

NewUser:$6$q1X1/ZB/NGu9ulrF$.JYhoCPsGT6hk0GD34oJYWwOKGxY67ka818lpy/0HY.7XvXK69Jd
eYOtMkNJNQvqBTTfblYBQ3SZ.MYxChPeQ1:1002:1002::0:0:John Doe,,,555 433423:/home/Ne wUser:/bin/tcsh

Is this the entry you wish to remove? Yes
```

3. Remove the user account with `yes`.

```
Remove user's home directory (/home/NewUser)? Yes
Removing user (NewUser): mailspool home passwd.
Administrator@CX-3B151A:~ %
```

4. Remove the home directory with `yes`.

⇨ The user account is deleted, along with the home directory. Entries from groups, temporary file storage areas and emails are removed. All processes initiated by the user are terminated.

7.3 Integrating a USB stick

This section shows you how you can integrate a USB stick in TwinCAT/BSD. Check the USB configuration beforehand. To do this, use the command `dmesg` to check whether the USB stick appears in the system messages:

```
umass0: <Generic Mass Storage, class 0/0, rev 2.00/1.01, addr 2> on usbus0
umass0: SCSI over Bulk-Only; quirks = 0x4101
umass0:1:0: Attached to scbus1
da0 at umass-sim0 bus 0 scbus1 target 0 lun 0
da0: <Generic Flash Disk 8.07> Removable Direct Access SPC-2 SCSI device
da0: Serial Number 2EFBC899
da0: 40.000MB/s transfers
da0: 3900MB (7987200 512 byte sectors)
da0: quirks=0x2<NO_6_BYTE>
```

The details of the make, device file (da0), speed and capacity may vary depending on the device.

Requirements:

- The USB stick is formatted with FAT32.
- Connect the USB stick to the Industrial PC.

Proceed as follows:

1. Enter the command `gpart show` to search for a FAT partition on the USB stick. The USB stick in this example has a FAT32 partition and a size of 3.8 GB. See entry: `1 fat32 (3.8G)`

   ```
   Administrator@CX-3B151A$ gpart show
   =>      40  7728256  ada0  GPT  (3.7G)
           40   409600     1  efi  (200M)
      409640     2008        - free - (1.0M)
      411648  7315456     2  freebsd-zfs (3.5G)
     7727104     1192        - free - (596K)

   =>      63  7987137  da0   MBR (3.8G)
           63      737        - free - (369K)
          800  7986400    1   fat32    (3.8G)
   Administrator@CX-3B151A$
   ```

2. Enter the command `ls /dev/da0*` to determine the name of the FAT32 partition. `da0s1` or `da0p1`.

   ```
   ls /dev/da0s1
   /dev/da0 /dev/da0s1
   ```

3. Enter the command `doas mkdir /mnt/usb` to create the directory `/mnt/usb`.
4. Enter the command `doas mount -t msdosfs /dev/da0s1 /mnt/usb` in order to mount the USB flash drive under `/mnt/usb`.
5. Use the command `cd /mnt/usb` to navigate to the mount point of the USB flash drive.
6. Use the command `ls` to display all the folders in the directory.

⇨ You have successfully integrated a USB stick in TwinCAT/BSD. The settings are not permanently saved. Following a restart the USB stick must be integrated again.

USB devices can also be integrated automatically. This function was disabled ex works in order to increase the safety of TwinCAT/BSD. Further information on the automount service can be found at: https://www.freebsd.org/doc/handbook/usb-disks.html

7.4 Disable real-time Ethernet

This chapter demonstrates how to disable real-time Ethernet if you do not need real-time communication and want to use the interfaces instead for traditional full-bandwidth Ethernet communication.

Real-time Ethernet is enabled by default. Use the command `TcRteConfig show` to call up the current configuration for the interfaces:

```
Administrator@CX-3B151A:~ % TcRteConfig show
sysctl:
dev.igb.1.iflib.tc_rte.mode: 0
dev.igb.1.iflib.support_tc_rte: 1
dev.igb.1.iflib.disable_tc_rte: 0
dev.igb.0.iflib.tc_rte.mode: 0
dev.igb.0.iflib.support_tc_rte: 1
dev.igb.0.iflib.disable_tc_rte: 0
```

Depending on the device, different network adapters can be displayed. In the example shown here, the network adapters are called `igb0` and `igb1`.

Disable real-time Ethernet as follows:

1. Enter the command `doas TcRteConfig disable igb.1` in the console.
   Real-time Ethernet is disabled for the `igb.1` interface.

   ```
   /boot/device.hints:
   dev.igb.1.iflib.disable_tc_rte="1"
   Administrator@CX-3B151A:~ %
   ```

2. Restart the Industrial PC with the command `shutdown -r now` to apply the settings.

⇨ After the restart, real-time Ethernet is disabled for the `igb.1` interface. The command `doas TcRteConfig enable igb.1` can be used to re-enable real-time Ethernet. Here, too, the settings are only applied after a restart.

7.5 Start services automatically (Autostart)

This section shows how to start services automatically after booting. A suitable entry in the *rc.conf* file is required under TwinCAT/BSD. The Mosquitto MQTT broker is shown as an example.

The *rc.conf* file generally contains information about the system configuration such as the local host name, configuration details for possible network interfaces and which services should be started when the system starts up.

Requirements:

- Mosquitto MQTT broker.

Proceed as follows:

1. Enter the command `doas ee /etc/rc.conf` on the console.
   The file rc.conf is opened.
2. In the editor navigate to the end of the file and create the following entry:
   `mosquitto_enable="YES"`
3. Press **[Esc]** and save the changes.

⇨ The Mosquitto MQTT broker is automatically started at the next system start.

7.6 Changing the shell

A shell is a command line interface that enables the user to interact with TwinCAT/BSD and execute commands. TwinCAT/BSD already contains some shells at delivery, including the `sh shell` set as standard. The user can switch to another shell at any time, for example the tcsh shell, or install additional shells.

The shells differ in the built-in functions or are preferred by users if they make their daily work easier, for example by auto-completion of file names. Which shell is used by the user is in the end a question of personal preference.

All existing shells in the system are listed under `/etc/shells`.

Proceed as follows:

1. Enter the command `chsh -s tcsh` in the console to switch to the tcsh shell.
2. Log in again with the command `login <username>`.

⇨ The tcsh shell is now selected as default shell for the logged in user. You can change the shell again at any time with `chsh -s`.

7.7 Change keyboard language

If you want to change the keyboard language, the quickest way is to use the command `kbdmap` and it applies to the current session. Create an entry in the file `rc.conf`, so that the language settings remain even after a restart.

Proceed as follows:

1. Open the file `rc.conf` with the command `doas ee /etc/rc.conf`
2. Navigate with the arrow keys to the end of the file and add the following line:
   `keymap="de.kbd"`.
3. Use an appropriate country code. In this example, `de` is used as the country code for German.

⇨ The language settings in the file `rc.conf` are permanently applied. Restart TwinCAT/BSD with `doas shutdown -r now` for the changes to take effect. Remove the entry to restore the default settings.

7.8 Synchronize time with NTP

The internal time of an Industrial PC is never completely accurate, which is why there is a way to determine and set the exact time with the Network Time Protocol (NTP). To get the current time, an NTP client is used to synchronize the local system time with a time server. For this purpose, Beckhoff offers a global NTP server pool via which the current time is made available:

```
ntp.beckhoff-cloud.com
```

TwinCAT/BSD is preconfigured to use this NTP server pool to determine the current time. The configuration of the NTP client is done via the configuration file:

```
/etc/ntp.conf
```

There you will find the entry for the Beckhoff NTP server pool `ntp.beckhoff-cloud.com`. If you want to use your own NTP server, replace the entry with the address of your NTP server.

These settings are alternatively possible via the web interface of the Beckhoff Device Manager (see: Beckhoff Device Manager: web interface [▶ 42]).

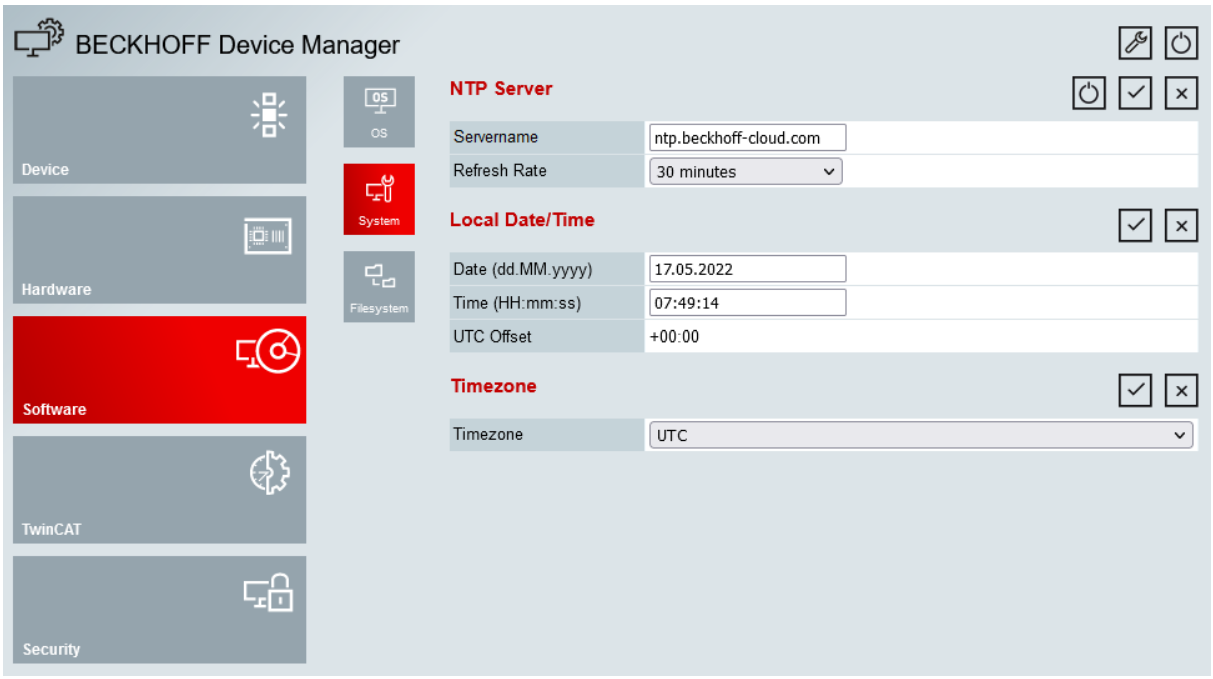


Fig. 6: NTP server settings in the Beckhoff Device Manager.

Beckhoff NTP server pool

The Beckhoff NTP server pool is a global server pool with several time servers per geographical region. To use only the best available servers for your region, simply use the global address `ntp.beckhoff-cloud.com` and the right servers for your region will be automatically made available.

This service is only available for Beckhoff Industrial PCs and may only be used elsewhere with the express permission of Beckhoff. Beckhoff does not guarantee the continuous availability of the time servers. Accordingly, Beckhoff assumes no liability for a time server failure.

https://www.beckhoff.com/ntp-pool

8 Remote access

8.1 Beckhoff Device Manager: web interface

With the Beckhoff Device Manager, Beckhoff Industrial PCs can be diagnosed and important system values monitored in order to avoid downtimes. With a sophisticated system diagnostics, it is thus possible to detect critical conditions at an early stage, such as an impending heat collapse due to the failure of a fan or insufficient cooling in the control cabinet. The Beckhoff Device Manager is only available for Beckhoff Industrial PCs, since it requires a customized BIOS.

Both remote access from another PC and access from a PLC to the Beckhoff Device Manager are supported. This allows the Device Manager to be configured easily and intuitively.

Web interface

The web interface can be accessed via a standard web browser. To do this, enter the IP address or the host name of the Industrial PC in the search bar.

- Example with IP address: https://169.254.136.237
- Example with host name: https://CX-16C2B8

Fig. 7: Start page of the Beckhoff Device Manager.

On the start page of the Beckhoff Device Manager you can either start the Device Manager or the TwinCAT/BSD web console. The web console gives you access to the TwinCAT/BSD console, allowing you to operate the Industrial PC without a monitor.

Access to the web interface is protected by the login data and the default password of the administrator. You should change this password to prevent unauthorized access to the system. Access data on delivery:

- User name: `Administrator`
- Password: `1`

The host PC and the Industrial PC must be in the same network and the firewall must allow access via port 443 (HTTPS). Port 443 is enabled ex works. Depending on the structure and configuration of your network (proxy server, etc.), the host name may not be resolved. We therefore recommend using the IP address of the Industrial PC.

Device Manager: first page

The Beckhoff Device Manager is started after login. The first page provides a basic overview of the device. From here you can directly access the hardware, software, TwinCAT and security sections. This allows you to check the hardware and software in a targeted manner.

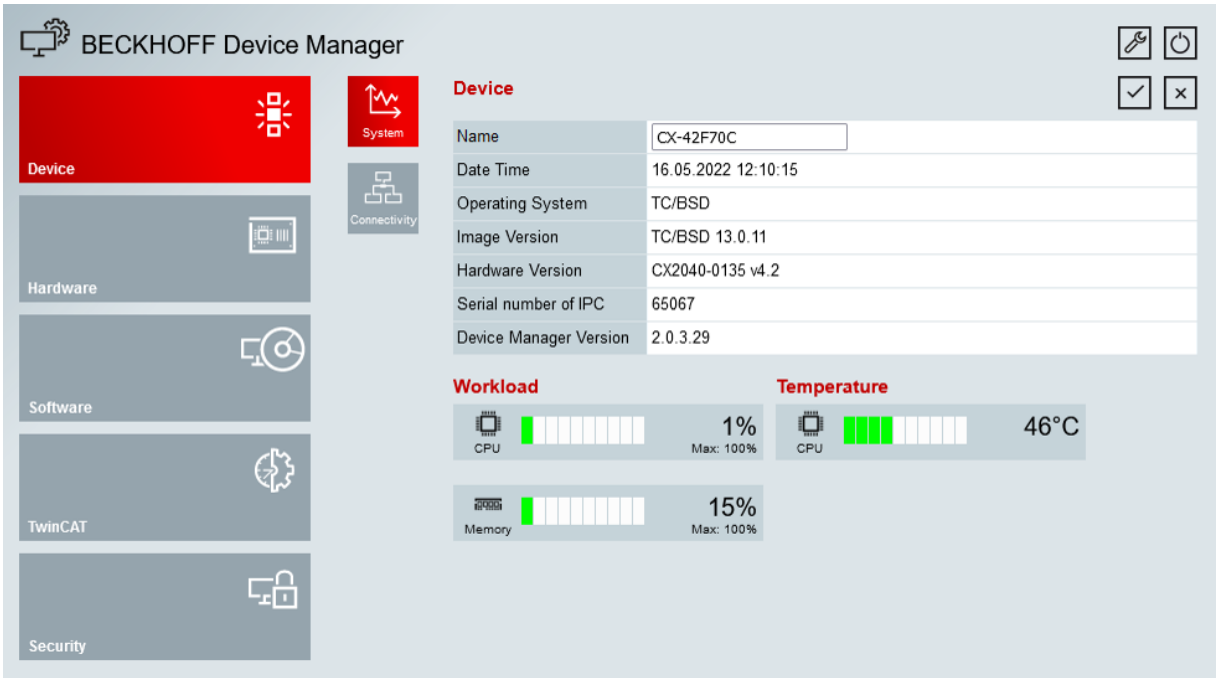


Fig. 8: First page of the Beckhoff Device Manager.

From the home page, navigate further in the menu to configure the Industrial PC. Note that modifications only become active once they have been confirmed. It may be necessary to restart the Industrial PC.

Setting options

The following information and settings are available for TwinCAT/BSD:

- General device information: listing of the TwinCAT/BSD and the TwinCAT version. In addition to this, information about the load of the CPU and the main memory is displayed.
- Network settings: the network settings can be changed independently of the console and, for example, the IP addresses of the Ethernet interfaces can be changed or DHCP can be enabled or disabled.
- Storage media and file system: both the free storage space and the lifetime of the storage media are displayed.
- TwinCAT/BSD packages: listing of the installed Beckhoff packages with version information, which are either relevant for TwinCAT or the TwinCAT/BSD operating system.
- TwinCAT: in the TwinCAT menu ADS routes can be managed and new ADS routes can be created.
- User management: new users or user groups can be created.

8.2 Remote access with the PuTTY client

PuTTY is an open source software with which a connection can be established via Secure Shell (SSH), Telnet, remote login or a serial interface.

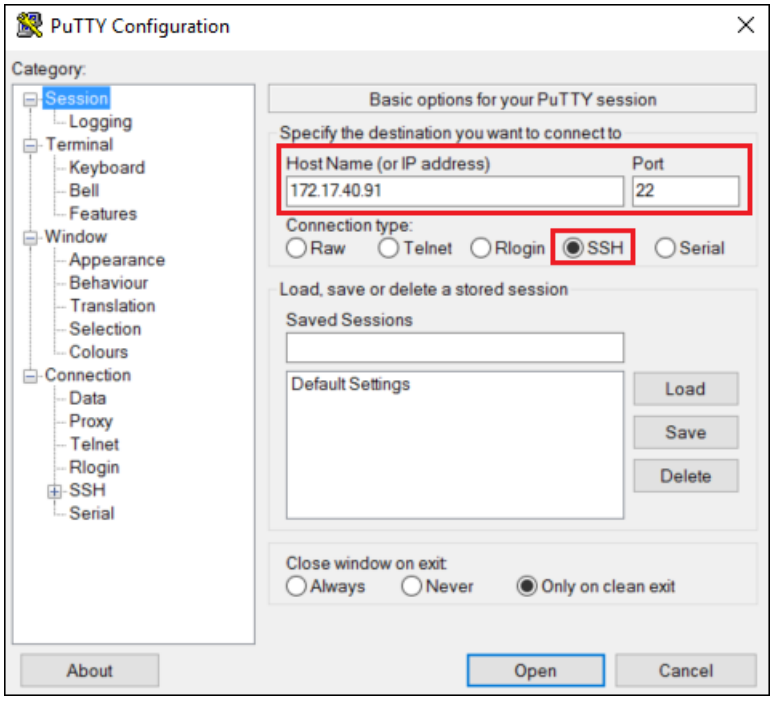
Use PuTTY to establish an SSH connection to an Industrial PC with TwinCAT/BSD when running Windows. After successful connection a console is started with which remote access commands can be issued that are subsequently executed on the Industrial PC.

Requirements:

- Download the PuTTY client from: https://www.putty.org/
  Minimum version required: 0.70
- The local PC (development computer) and the Industrial PC (TwinCAT/BSD) must be connected the same network or directly to each other via an Ethernet cable.

Start an SSH connection as follows:

1. Start the PuTTY client.
2. Enter the host name or the IP address of the Industrial PC in **Host Name (or IP address)**.



3. Enter the appropriate port in **Port**. For SSH this is usually port 22.
4. Activate the option **SSH** in **Connection type** and click on **Open**.
   The console is started.

```
login as:
```

5. Enter the login data for TwinCAT/BSD. Standard:
   Login: `Administrator`
   Password: `1`

⇨ You have successfully established an SSH connection to an Industrial PC with TwinCAT/BSD using the PuTTY client.

8.3 Managing files with the WinSCP client

8.3.1 Starting and using the WinSCP client

WinSCP (Windows Secure Copy) is an open source software with which a connection can be established via FTP, FTPS, SCP, SFTP, WebDAV or S3h.

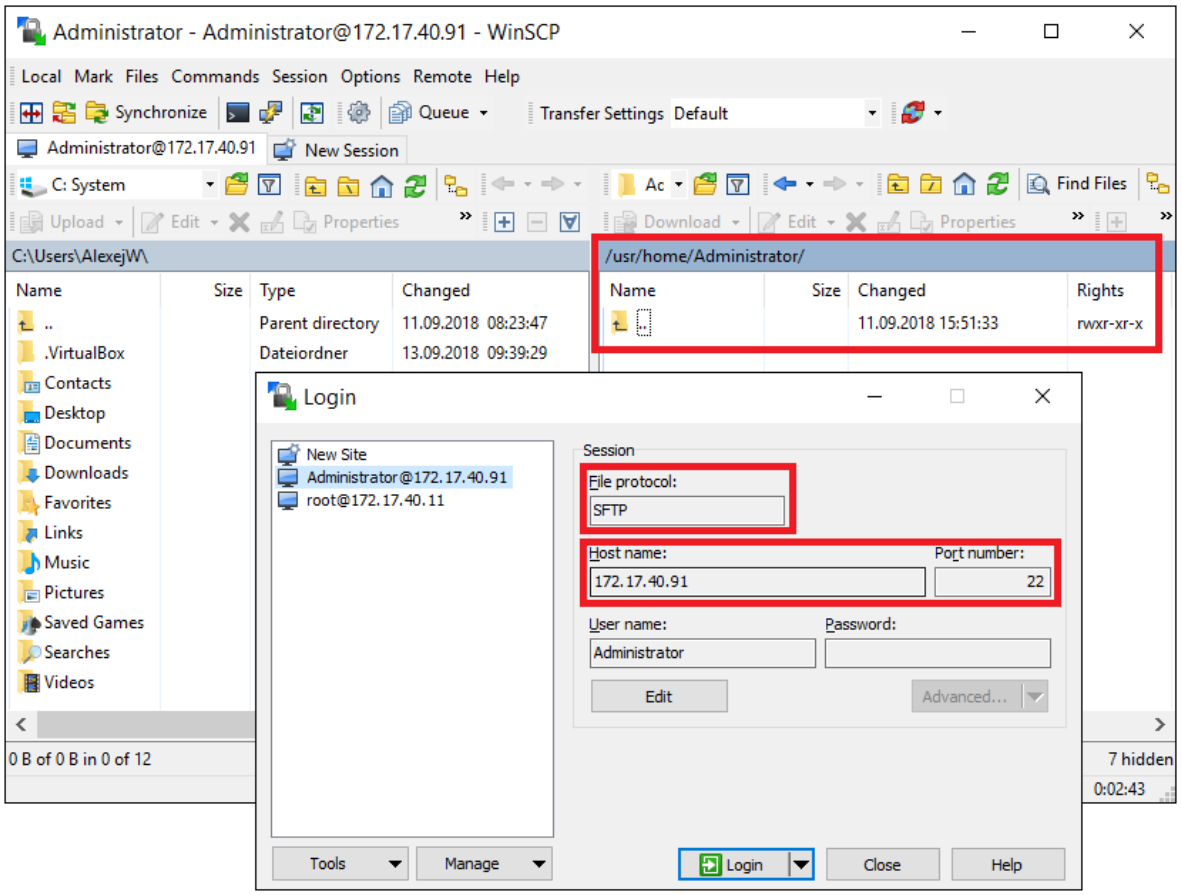
Use WinSCP to establish an SFTP connection to an Industrial PC with TwinCAT/BSD when running Windows. Following successful connection, a graphical user interface is started with which it is possible to securely transfer data and files to the Industrial PC with TwinCAT/BSD. With WinSCP, files can be copied into TwinCAT/BSD directories, new directories can be created and files can be edited.

Requirements:

- Download WinSCP from: https://winscp.net/
  Minimum version required: 5.13.2
- The local PC (development computer) and the Industrial PC (TwinCAT/BSD) must be connected the same network or directly to each other via an Ethernet cable.

Start a connection as follows:

1. Start the WinSCP client.
   The login window appears.



2. Select the SFTP protocol in **File protocol**.
3. Enter the IP address and the port number of the Industrial PC in **Host Name** and **Port number**.
4. Enter the login data for TwinCAT/BSD and click on **Login**.

⇨ You have successfully established an SFTP connection to an Industrial PC with TwinCAT/BSD and can securely transfer data and files to the Industrial PC. The TwinCAT/BSD directories are shown on the right-hand side of the graphical user interface.

8.3.2 WinSCP as the root

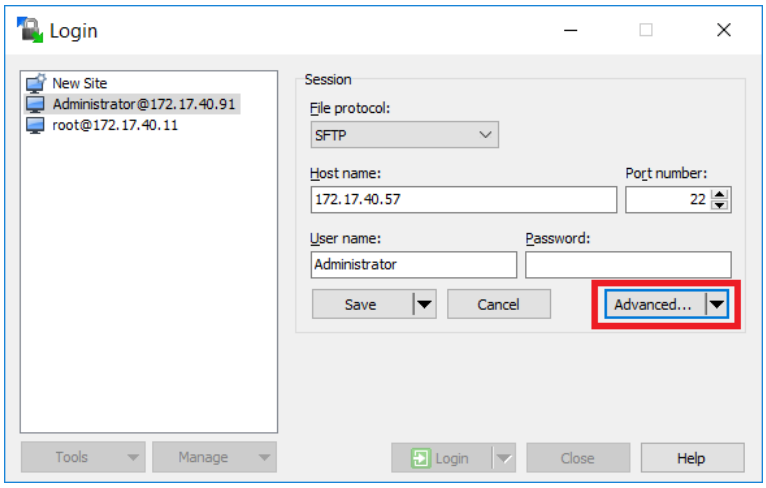
With the entry `doas /usr/libexec/sftp-server`, WinSCP starts the SFTP server with root rights. This allows additional settings to be made and configuration files to be adapted.

Requirements:

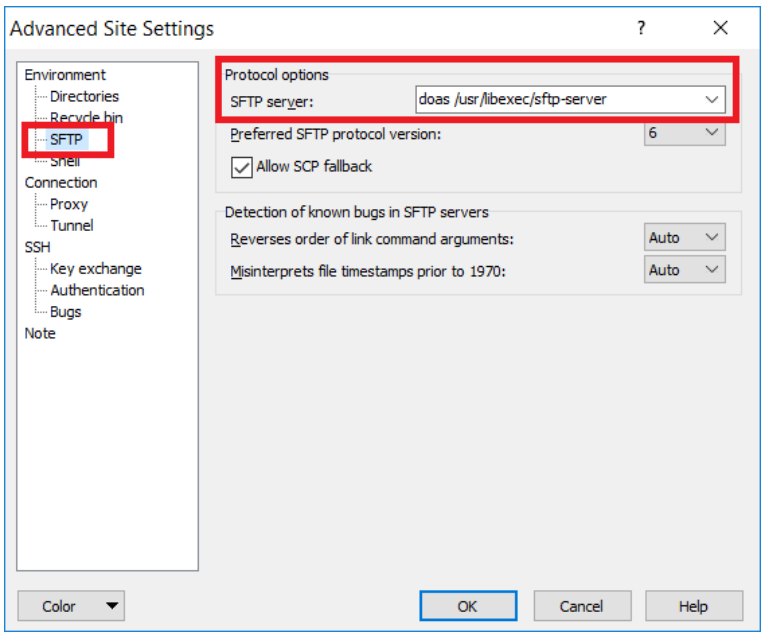
- WinSCP client has been installed.

Proceed as follows:

1. Start the WinSCP client.
   The login window appears.
2. Click on **Advanced** to open further settings.



3. Click on **SFTP** in the tree view on the left and enter the value *doas /usr/libexec/sftp-server* in **SFTP server**.



4. Save the settings for the administrator account.

⇨ Then log in with the administrator account. You now have access with root rights via WinSCP.

8.3.3 Opening and editing files

With the WinSCP client you can open and edit files with the help of a graphical interface. Please note that you can only edit files for which you have the required access rights.

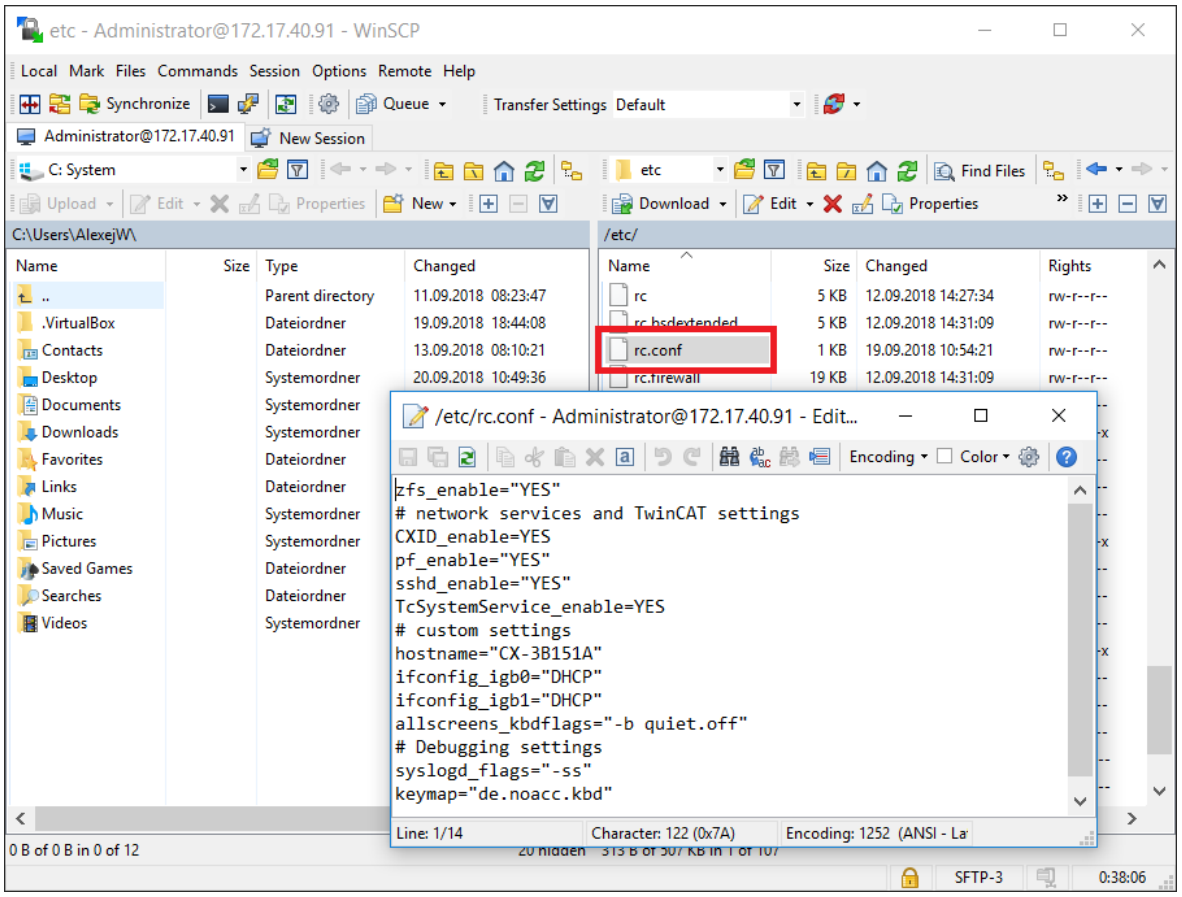
Taking the configuration file `rc.conf` as an example, this work step shows how files can be opened and edited with the WinSCP client.

Requirements:

- WinSCP-Client (see: Starting and using the WinSCP client [▶ 45]).
- Activate root rights for WinSCP (see: WinSCP as the root [▶ 46]).

Proceed as follows:

1. Start the WinSCP client.
   The login window appears.
2. Enter the login data for TwinCAT/BSD and click on **Login**.
3. Navigate to the directory */etc* and double-click on the file *rc.conf*.
   The file is opened in the WinSCP editor.



4. Alternatively you can right-click on the file and open it with the editor of your choice.
5. As soon as you save changes, the changes will be transferred to TwinCAT/BSD.

⇨ You have successfully opened and edited a file. In this way you can manage all files with the WinSCP client.

8.4 Editing the SSH settings

SSH is restrictively configured in TwinCAT/BSD. Current encryption methods are used. If you experience problems in establishing a connection via SSH, you can edit the SSH settings and comment out restrictive SSH settings.

Note that in doing so the restrictive settings from Beckhoff for a secure network connection will be canceled. Beckhoff recommends the use of a different software for an SSH connection to the TwinCAT/BSD or to update the existing software.

Requirements:

- Access rights to the file *sshd_config*

Proceed as follows:

1. Enter the command `doas ee /etc/ssh/sshd_config` in the console.
   The file *sshd_config* is opened.
2. Comment out the following four lines to cancel the restrictive SSH settings.

```
#Ciphers chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-  ctr,aes128-
gcm@openssh.com,aes256-gcm@openssh.com
#HostKeyAlgorithms ssh-rsa,rsa-sha2-256,rsa-sha2-512,ssh-ed25519
#KexAlgorithms diffie-hellman-group14-sha256,diffie-hellman-group16- sha512,diffie-hellman-
group18-sha512,curve25519-sha256,curve25519- sha256@libssh.org
#MACs hmac-sha2-256-etm@openssh.com,hmac-sha2-512- etm@openssh.com,umac-128-etm@openssh.com
```

3. Restart the SSH server with the command `doas service sshd restart` in order to confirm the settings.

⇨ You can restore the restrictive SSH settings at any time by removing the comments again.

9 TwinCAT/BSD Hypervisor

The TwinCAT/BSD Hypervisor enables virtual machines to run under TwinCAT/BSD.

The TwinCAT/BSD Hypervisor is based on the FreeBSD hypervisor bhyve(4). The integration of `bhyve` in TwinCAT/BSD enables simultaneous and efficient execution of virtual machines and TwinCAT PLC and motion applications on the same industrial PC.

This documentation provides an overview of various features of the TwinCAT/BSD Hypervisor.

9.1 Device and feature support

Running virtual machines with bhyve(8) requires an industrial PC with a current Intel®- or AMD™ CPU that supports hardware-assisted virtualization.

The table Device support for TwinCAT/BSD Hypervisor, device and GPU passthrough. [▶ 49] gives an overview of current industrial PCs that enable the execution of virtual machines with the TwinCAT/BSD Hypervisor and whether the Device passthrough or GPU passthrough features are supported

Table 3: Device support for TwinCAT/BSD Hypervisor, device and GPU passthrough.

TwinCAT/BSD devices	Hypervisor	Device passthrough	GPU passthrough
CX51x0	Yes	No	No
CX52x0	Yes	Yes	No
CX20x2	Yes	Yes	No
CX20x3	Yes	Yes	No
C601x-0010	Yes	No	No
C601x-0020	Yes	Yes	No
C602x-0000	Yes	Yes	No
C603x-0060	Yes	Yes	No
C603x-0070	Yes	Yes	No

9.2 Start and manage virtual machines

Virtual machines are started and managed by the programs `bhyve` and `bhyvectl`. Before a virtual machine can be started with `bhyve`, the kernel module `vmm.ko` must be loaded:

```
doas kldload -n vmm.ko
```

So that this step does not have to be repeated after each restart, the kernel module can already be loaded during the system startup of TwinCAT/BSD by setting `vmm_load="YES"` in the `/boot/loader.conf`:

```
doas sysrc -f /boot/loader.conf vmm_load="YES"
```

Once the kernel module is loaded, a virtual machine can be started by calling `bhyve`:

```
bhyve [OPTIONS] <vm_instance>
```

The parameters `[OPTIONS]` determine the configuration of the virtual machine, which can be used, for example, to specify the number of virtual CPUs used, the size of the main memory or the storage location. The last parameter `<vm_instance>` of the call specifies the virtual machine instance name.

Starting a VM instance with simple base configuration

A UEFI-based virtual machine with two virtual CPUs and 2 GB of main memory can be started with the following command:

```
doas bhyve \
-c sockets=1,cores=2,threads=1 \
-m 2G \
-s 0,hostbridge \
```

```
-s 31,lpc \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-l com1,stdio \
-A -H -P \
samplevm
```

In this sample, the virtual machine is started with the instance name `samplevm`. The UEFI output in this configuration is output through the standard streams on the command line as follows:

```
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (BHYVE, 0x00010000)
map: No mapping found.
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell>
```

By entering `reset -s` in the UEFI shell, the virtual machine can be shut down again. The bhyve process thereby terminates with the return value `1`.

To restart the virtual machine, the complete bhyve command must be called with the same parameters.

Explanation of the parameters

The meaning of the individual parameters can be called via `bhyve -h`. Detailed descriptions of the parameters can be found in the manual for bhyve. Alternatively, the manual can be accessed via the command line using the `man bhyve` command.

In the following, the parameters used in the above sample will be briefly explained.

Parameter	Description
`-c sockets=1,cores=2, threads=1`	Configuration of the virtual CPU topology. In this sample, a CPU socket with two cores and one thread per core.
`-m 2G`	Main memory available to the virtual machine. In this sample 2 GB.
`-s 0,hostbridge`	A virtual host bridge to connect the virtual CPU to the virtual PCI bus. By convention, the host bridge should always be configured at PCI address -s 0:0:0 (-s 0 for short).
`-s 31,lcp`	LPC/PCI ISA bridge for connecting emulated LPC devices. By convention, the LPC/PCI ISA bridge should always be configured at PCI address -s 0:31:0 (-s 31 for short).
`-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd`	An emulated bootrom on the LPC bus. The UEFI firmware is passed as ROM in the file `/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd`.
`-l com1,stdio`	A serial interface on the LPC bus whose inputs and outputs are redirected to bhyve's standard input and output streams.
`-A`	Creates bhyve ACPI tables for the virtual machine.
`-H`	Releases the virtual CPU thread when an HLT instruction is detected. Otherwise, the virtual CPUs will use 100% of the host CPUs.
`-P`	Forces the virtual guest CPU to terminate when a PAUSE instruction is detected.

Parameters starting with `-s` are used to configure virtual PCI slots to which in turn emulated PCI devices can be assigned (for examples see: Advanced VM configuration)

Parameters starting with `-l` are used to configure emulated LPC devices behind the LPC/PCI-ISA bridge.

Manage virtual machines

Started virtual machines are listed as bhyve processes on the TwinCAT/BSD host. Accordingly, `ps(1)` can be used to list running virtual machines:

```
ps -a | grep bhyve
```

```
7048 0 SC 0:31.06 bhyve: samplevm (bhyve)
7642 0 SC 0:01.83 bhyve: debian11 (bhyve)
```

Running virtual machines can be shut down or terminated via the TwinCAT/BSD host by sending signals via `kill(1)` to the respective bhyve process. The TERM signal can be used to send an ACPI shutdown request to the virtual machine to trigger the virtual machine shutdown:

```
doas kill -s TERM $(pgrep -f "bhyve: samplevm")
```

If the virtual machine does not respond to ACPI shutdown requests, the KILL signal can be used to terminate the bhyve process directly:

```
doas kill -s KILL $(pgrep -f "bhyve: samplevm")
```

After a bhyve process has been terminated, its return value (exit code) can be queried by the shell variable $?:

```
echo $?
```

Return values greater than 1 indicate that the virtual machine could not be shut down properly. If the virtual machine is to be restarted or if the configuration of a VM instance is changed between bhyve calls, the VM instance must first be removed via `bhyvectl :`

```
doas bhyvectl --vm=samplevm --destroy
```

Virtual machine instances are listed as device files at `/dev/vmm` and can be further managed using `bhyvectl`. Via `ls -al /dev/vmm` it can also be determined which virtual machines are currently created on the TwinCAT/BSD host:

```
ls /dev/vmm
debian11 samplevm
```

9.3 Use shell scripts

Starting and managing virtual machines can be simplified and automated by using shell scripts. Scripted VM applications can be used to persistently store configurations that can be reused even after a restart. In combination with further instructions and shell scripts, any VM applications can thus be set up under TwinCAT/BSD.

The following sample script shows the basic structure of a scripted VM application:

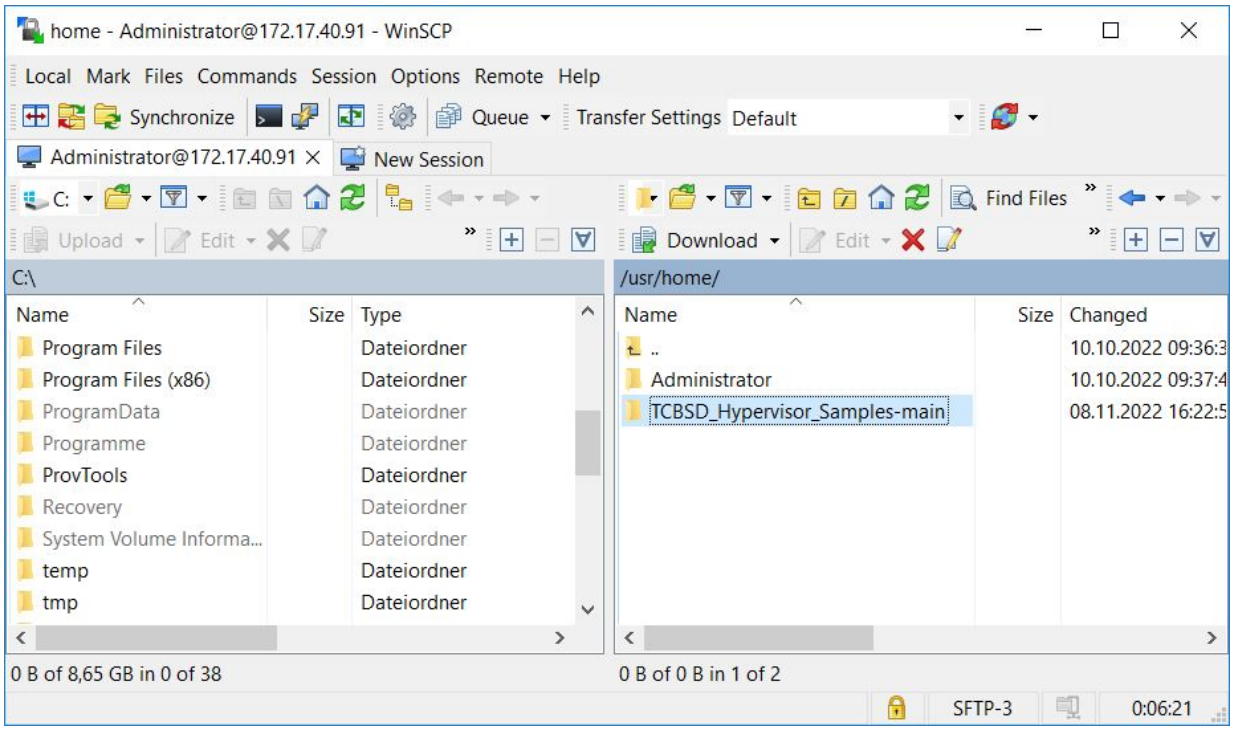
```
# root permissions are required to run VMs
if test "$(id -u)" -ne 0; then
printf "%s must be run as root\n" "${0##*/}"
exit 1
fi

# Default values for VM configuration
vm_name="samplevm"

# Ensure that kernel modul vmm.ko is loaded
kldload -n vmm.ko

while true; do
# destroy former VM instance to ensure we start
# with a clean VM configuration
if test -e "/dev/vmm/${vm_name}"; then
bhyvectl --vm="${vm_name}" --destroy
fi

# start a simple UEFI based VM instance
_bhyve_rc=0
bhyve \
-A -H -P \
-c sockets=1,cores=1,threads=1 \
-m 1G \
-s 0:0,hostbridge \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-l com1,stdio \
-s 31:0,lpc \
"${vm_name}"
_bhyve_rc=$?

# according to bhyve man pages the return codes indicates
# how the VM was terminated:
# 0: rebooted
# 1: powered off
# ...
# 4: exited due to an error
if test "${_bhyve_rc}" -ne 0; then
printf "bhyve exited with return code: %s\n" "${_bhyve_rc}"
break
fi
printf "Restarting %s\n" "${vm_name}"

done
```

The sample script can be saved to a text file on the TwinCAT/BSD host and executed. Alternatively, the sample script can be downloaded from GitHub repository at https://github.com/Beckhoff/TCBSD_Hypervisor_Samples and copied to the TwinCAT/BSD host.

Proceed as follows:

1. Download the sample script at https://github.com/Beckhoff/TCBSD_Hypervisor_Samples.

2. Copy the entire folder on the TwinCAT/BSD host to the directory */usr/home* by using the WinSCP program, for example.



3. Navigate to the new directory with `cd /usr/home/TCBSD_Hypervisor_Samples-main/basic_vm_script`.
4. Enter the command `doas make` to install the sample script `samplevm`. In addition to the installation, the file permissions are set, making the sample script executable. Without the `doas make` command, file permissions must be set manually to run the sample script.
5. Finally, enter the command `doas samplevm` to run the sample script.

⇨ The virtual machine boots into the UEFI shell, which is output to the command line.

```
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (BHYVE, 0x00010000)
map: No mapping found.
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell>
```

You can return to the command line by shutting down the virtual machine with the `reset -s` command. To start virtual machines as a system service under TwinCAT/BSD, shell scripts can be used in combination with the rc framework and thus virtual machines can be managed as a system service or started automatically at system startup (see: Autostart shell scripts [▶ 54]).

9.4 Autostart shell scripts

To manage a virtual machine as a system service or to start it automatically at system startup, the call of bhyve or a shell script can be included in the rc framework.

The GitHub repository: https://github.com/Beckhoff/TCBSD_Hypervisor_Samples/tree/main/vm_autostart includes sample files that demonstrate how to incorporate a VM configuration into the rc framework using shell scripts. The sample in the directory `vm_autostart` contains the appropriate files for this:

```
vm_autostart
├── Makefile
├── rc.d
│   └── samplevm
└── samplevm
```

The sample script `samplevm` shown in the chapter Use shell scripts [▶ 52] has been extended by `start`, `stop` and `status` parameters to be able to start and stop a VM configuration with VNC access via the command line.

The `vm_autostart/rc.d/samplevm` shell script is used to integrate the `vm_autostart/samplevm` shell script into the rc framework.

Controlling the VM system service with the sample script:

1. Navigate to the directory with *cd /usr/home/TCBSD_Hypervisor_Samples-main/vm_autostart*
2. Enter the command `doas make` to install both files from the `vm_autostart` directory on the TwinCAT/BSD host.
3. Then enter `doas service samplevm enable` to enable the VM instance as a system service for autostart via `service(8)`.
4. After including the shell script as a system service, the virtual machine can be started with the `doas service samplevm start` command.

⇨ In this sample, the virtual machine can be accessed via a VNC client on TCP port "5900". From now on, the virtual machine is also restarted after a reboot of the TwinCAT/BSD host and is available for use. The virtual machine can be stopped again with the command `doas service samplevm stop`.

The sample script is a first starting point and illustrates how virtual machines can be started, managed and automated under TwinCAT/BSD. The sample script `vm_autostart/samplevm` can be customized and extended as needed to achieve a desired VM configuration. The chapter Advanced VM configuration explains other parameters that can be used to extend the configuration of a virtual machine.

For detailed information about creating rc.d scripts, see the FreeBSD Handbook chapter Practical rc.d scripting in BSD.

9.5 UEFI-based virtual machines

UEFI-based virtual machines can be started by the parameters `-l,bootrom,<efi-rom>[,<efi-vars>]`. For `<efi-rom>`, the path to an EFI ROM file must be specified. Optionally, the path to a file can be specified at `<efi-vars>`, which in turn serves as a location for EFI virtual machine variables.

Files with EFI variables should be created per VM instance. The following command creates a copy of the `BHYVE_BHF_UEFI_VARS.fd` file to be used for the `samplevm` VM instance.

```
doas cp /usr/local/share/uefi-firmware/BHYVE_BHF_UEFI_VARS.fd /vms/samplevm/EFI_VARS.fd
```

The `EFI_VARS.fd` file is then passed as `<efi-vars>` parameter to the `bhyve` call:

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd,/vms/samplevm/EFI_VARS.fd \
-l com1,stdio \
-s 0:0,hostbridge \
-s 31:0,lpc \
-A -H -P \
samplevm
```

9.6 VNC-based interaction with virtual machines

NOTE

Unsecured TCP port

Incoming connections on TCP port 5900 are not blocked by the firewall in this sample. Set up a secure and encrypted connection and secure TCP port 5900 via SSH as soon as the operation takes place in an unsecured network.

With Virtual Network Computing (VNC) it is possible to control a virtual machine on a TwinCAT/BSD host via a network connection. For this, `bhyve` provides an integrated VNC server to interact with VM instances.

Virtual machine graphical output and user input to the virtual machine can be transmitted via the integrated VNC server by configuring the virtual machine with a frame buffer device `fbuf`. The following options can be passed to the frame buffer device `fbuf` to configure the VNC server:

```
fbuf,[rfb=ip-and-port][,w=width][,h=height][,vga=vgaconf][,wait][,password=password]
```

The following call starts the virtual machine `samplevm` with a frame buffer device at PCI slot 2. The configuration options specify that the VNC server listens for connections on TCP port 5900 of the TwinCAT/BSD host. In addition, the image resolution of the frame buffer is set to 1024x768 pixels. Further configuration options of the `fbuf` device can be found in the bhyve man pages

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 2,fbuf,rfb=0.0.0.0:5900,w=1024,h=768 \
-s 31,lpc \
-A -H -P \
samplevm
```

Depending on the VNC client, mouse pointer positions may not be passed accurately. The bhyve call can then be extended to include a `xhci,tablet` device configuration.

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 2,fbuf,rfb=0.0.0.0:5900,w=1024,h=768 \
-s 3,xhci,tablet \
-s 31,lpc \
-A -H -P \
samplevm
```

The integrated VNC server does not support transport layer security. Incoming TCP connections on port 5900 are blocked by default by the TwinCAT/BSD packet filter `pf(8)`. Incoming connections can be allowed by configuring the packet filter (see: Firewall).

9.7 ZFS data sets as storage location for virtual machines

Using ZFS data sets offer the possibility to use functions and properties of ZFS like quotas, compression, block sizes or snapshots for VM applications.

The following call creates a file system as location for the virtual machine `samplevm` and mounts the file system in the directory structure at `/vms/samplevm`:

```
doas zfs create -p -o mountpoint=/vms/samplevm zroot/vms/samplevm
```

The file system can now be used to back up files for virtual drives, EFI variables, or other VM-related data.

Backup points from virtual hard disks via ZFS snapshots

ZFS snapshots can be applied to ZFS data sets to create backup points of virtual hard disks (see also: ZFS volumes as data memory for virtual hard disks [▶ 57]). The state of a virtual hard disk can thus be backed up at a specific point in time and restored if necessary.

A snapshot of the ZFS volume `zroot/vms/samplevm/disk0` can be created via zfs-snapshot(8):

```
doas zfs snapshot zroot/vms/samplevm/disk0@latest
```

`@latest` defines the name of the snapshot.

To restore the ZFS volume to the state of the snapshot `@latest`, the following command can be used

```
doas zfs rollback zroot/vms/samplevm/disk0@latest
```

During the build and restore process, the VM instance should be shut down.

For more detailed information about the Z file system and the use of ZFS volumes and snapshots, see the chapter The Z file system of the FreeBSD documentation

9.8 Virtual drives

Virtual machines can be configured with virtual drives (block storage devices). These can in turn be used as virtual hard disks (nvme, ahci-hd or virtio-blk) or as a virtual CD-ROM drive (ahci-cd).

The following call starts the virtual machine `samplevm` with an emulated NVMe drive and a virtual AHCI CD-ROM drive:

```
doas bhyve \
-c sockets=1,cores=2,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 2,fbuf,rfb=0.0.0.0:5900,w=1024,h=768 \
-s 3,xhci,tablet \
-s 10,nvme,/usr/home/Administrator/samplevm/disk0.img \
-s 15,ahci-cd,/usr/home/Administrator/samplevm/os-installation.iso,ro \
-s 31,lpc \
-H -P -A \
samplevm
```

Both drives use regular disk image files on the TwinCAT/BSD host as data memory, which must exist before bhyve is called. Alternatively, block devices such as ZFS volumes can be passed as data memory so that ZFS snapshots can be used to back up and restore virtual hard disk (see: ZFS data sets as storage location for virtual machines [▶ 56]).

In the sample above, the disk image file `/usr/home/Administrator/samplevm/disk0.img` is used as data memory for the virtual hard disk (see: Disk image files as data memory for virtual hard disks [▶ 57]).

The file `/usr/home/Administrator/samplevm/os-installation.iso` is only accessed for reading. If the memory image of `os-installation.iso` corresponds to a bootable ISO image, the installation of an operating system can be started within the virtual machine, for example (see: Installing a guest operating system using Debian Linux as an example [▶ 64]).

To make the configured drives known to guest operating systems via ACPI, the parameter `-A` must also be passed.

9.8.1 Disk image files as data memory for virtual hard disks

Disk image files are regular files in which the contents of virtual hard disks can be stored. An empty disk image file of maximum 20 GB can be created with the help of `truncate(1)` as follows:

```
truncate -s 20G disk0.img
```

The created `disk0.img` file can then be passed to the bhyve call as a backend for a virtual hard disk:

```
doas bhyve \
-c sockets=1,cores=2,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 10,nvme,disk0.img \
-s 31,lpc \
-H -P -A \
samplevm
```

9.8.2 ZFS volumes as data memory for virtual hard disks

ZFS volumes are a type of ZFS data sets and are listed as block devices at `/dev/zvol/zroot`. A ZFS volume can be used as data memory for virtual drives in order to take advantage of ZFS data sets such as snapshots, clones or compression.

The following command creates the ZFS volume `zroot/vms/samplevm/disk0` in the ZFS pool `zroot` with 20 GB

```
doas zfs create -V 20G zroot/vms/samplevm/disk0
```

The following call starts the `samplevm` virtual machine with an emulated NVME hard disk that uses the ZFS volume `zroot/vms/samplevm/disk0` as data storage, which is available under the directory `/dev/zvol/zroot/vms/samplevm/disk0`.

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 2,fbuf,rfb=0.0.0.0:5900,w=1024,h=768 \
-s 3,xhci,tablet \
-s 10,nvme,/dev/zvol/zroot/vms/samplevm/disk0 \
-s 31,lpc \
-A -H -P \
samplevm
```

9.8.3 Use of installation media (ISO images)

Installation programs for operating systems are often made available for download as ISO images via websites. Under TwinCAT/BSD `fetch(8)` can be used to download an ISO image from a website.

The following call loads the Debian ISO image `debian-11.5.0-amd64-netinst.iso` from the website cdimage.debian.org and saves it locally in the file `os-installer.iso`.

```
fetch -o os-installer.iso https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-11.5.0-amd64-netinst.iso
```

The downloaded ISO file can then in turn be used as media in a virtual CD-ROM drive (ahci-cd) of a virtual machine.

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 2,fbuf,rfb=0.0.0.0:5900,w=1024,h=768 \
-s 3,xhci,tablet \
-s 15,ahci-cd,/usr/home/Administrator/os-installer.iso,ro \
-s 31,lpc \
-A -H -P \
samplevm
```

9.9 Virtual machine network configuration

Virtual machines can be configured with virtual network controllers to connect the virtual machine to a network. Virtual machines use `tap(4)` or vmnet(4) devices of the TwinCAT/BSD host for this purpose, which in turn are managed under TwinCAT/BSD with the help of `ifconfig(8)`.

The following command creates a new `vmnet(4)` instance:

```
doas ifconfig vmnet create
```

```
vmnet0
```

To create the `vmnet0` instance already at system startup it can be added in the rc configuration `cloned_interfaces`:

```
doas sysrc cloned_interfaces+="vmnet0"
```

The created `vmnet(4)` instance (in this case `vmnet0`) can then be used as Ethernet endpoint for a virtual machine to exchange Ethernet packets between TwinCAT/BSD host and the virtual machine environment.

For this, the `bhyve` call is passed around an emulated `virtio-net` device on a PCI slot, which uses the previously created `vmnet0` interface as its endpoint. The following command starts a virtual machine with a virtual network controller that uses the previously created `vmnet0` device on the PCI slot `-s 20`:

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
```

```
-s 2,fbuf,rfb=0.0.0.0:5900,w=1024,h=768 \
-s 3,xhci,tablet \
-s 20,virtio-net,vmnet0 \
-s 31,lpc \
-A -H -P \
samplevm
```

Thus, virtual machines are always connected to external networks via `tap(4)` or `vmnet(4)` devices.

Depending on the use case, a virtual machine can also be configured with multiple network controllers. The connection of a virtual machine to a network is then determined by the configuration of the respective `tap(4)` or `vmnet(4)` devices on the TwinCAT/BSD host. This results in different possibilities to realize the communication of virtual machines into a network:

1. Host-Only network [▶ 59]
2. NAT network [▶ 59]
3. Bridged network [▶ 60]
4. Ethernet device passthrough [▶ 63]

9.9.1 Host-Only network

In a Host-Only network configuration, network packets are only exchanged between the virtual machine and the TwinCAT/BSD host. For this purpose, a `vmnet(4)` device with `ifconfig` is created on the TwinCAT/BSD host and a private IP address is assigned to this device.

```
doas ifconfig vmnet create inet 192.168.1.10 netmask 255.255.255.0
```

So that the vmnet instance is already created at system startup, it can be stored in the rc configuration:

```
doas sysrc cloned_interfaces+="vmnet0"
doas sysrc ifconfig_vmnet0="inet 192.168.1.10 netmask 255.255.255.0"
```

The virtual machine is then passed the vmnet(4) device as a backend for a virtual network controller (see: Virtual machine network configuration [▶ 58]).

The following call starts the virtual machine `samplevm` with a virtio-net based network controller at PCI address 20. The previously configured vmnet0 instance is passed as the backend.

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 20,virtio-net,vmnet0 \
-s 31,lpc \
-A -H -P \
samplevm
```

Within the guest system, the virtual network interface must be configured to be on the same IP network as the vmnet0 interface of the TwinCAT/BSD host (sample above: 192.168.1.0/24). Afterwards the Host-Only communication can be checked with ping requests between TwinCAT/BSD host and guest system.

9.9.2 NAT network

A NAT network can be used to send requests from a private VM network (for example, a Host-Only network) to an external network. Under TwinCAT/BSD the forwarding of IP packets between network interfaces must be activated for this:

```
doas sysctl net.inet.ip.forwarding=1
```

To save this setting persistently `net.inet.ip.forwarding=1` can be added to the file `/etc/sysctl.conf`. In addition, the translation of private network addresses to an external network requires appropriate network address translation (NAT) rules in pf(8).

The following sample uses the `vmnet0` configuration from chapter Host-Only network [▶ 59] for the private network between virtual machine and TwinCAT/BSD host.

```
ifconfig vmnet0
```

```
vmnet0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 58:9c:fc:10:56:5b
        inet 192.168.1.1 netmask 0xffffff00 broadcast 192.168.1.255
        groups: vmnet
        media: Ethernet autoselect
        status: no carrier
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
```

The IPC is connected to an external network via the physical network interface `igb0`:

```
ifconfig igb0
```

```
igb0: flags=8863<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=4a004a9<RXCSUM,VLAN_MTU,JUMBO_MTU,VLAN_HWCSUM,LRO,RXCSUM_IPV6,NOMAP>
        ether 00:01:05:62:3b:b0
        inet 172.17.98.154 netmask 0xffffff00 broadcast 172.17.98.255
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
```

For the translation of private addresses to the external network, the following pf rule can first be saved in a text file and then loaded via `pfctl(8)`:

```
nat on igb0 from vmnet0:network to any -> (igb0)
```

```
doas pfctl -a "bhf-nat/samplevm-nat" -f samplevm.nat.conf
```

Additionally, incoming network traffic should be allowed into the private network:

```
pass from vmnet0:network to any keep state
```

The rule set can in turn be saved in a text file and loaded via `pfctl(8)`:

```
doas pfctl -a "bhf/bhyve/samplevm-nat" -f samplevm.filters.conf
```

Once both rule sets are loaded, the VM can be started with `vmnet0` as the backend for the virtio-net based network controller:

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 20,virtio-net,vmnet0 \
-s 31,lpc \
-A -H -P \
samplevm
```

Within the guest operating system, communication into the external network can be checked with ping requests:

```
ping google.com
```

```
PING google.com (142.251.37.14): 56 data bytes
64 bytes from 142.251.37.14: icmp_seq=0 ttl=118 time=12.969 ms
64 bytes from 142.251.37.14: icmp_seq=1 ttl=118 time=12.817 ms
^C
```

It should be noted that the virtual network interface in the guest operating system is assigned a network address in the range of the `vmnet0` network (192.168.1.0/24 see above). In addition, the `vmnet0` address (192.168.1.1) must be entered as the default gateway, and addresses of name servers must be stored to resolve domain names such as google.com.

9.9.3 Bridged network

In a Bridge network, a physical network interface of the TwinCAT/BSD host (e.g. `igb0`) is connected to a tap(4) device via a bridge(4) device. The tap(4) device in turn serves as a backend for a network interface of the virtual machine (see: Virtual machine network configuration [▶ 58]).

Network communication of the virtual machine is thus bridged to the physical network interface of the TwinCAT/BSD host via the respective tap(4) device by a bridge(4) device on Ethernet level.

Configuration of the network components

A bridge(4) device is created on the TwinCAT/BSD host using ifconfig(8):

```
doas ifconfig bridge create
```

The output `bridge0` appears.

Likewise, a tap(4) device is created as the backend for the virtual network interface:

```
doas ifconfig tap create
```

The output `tap0` appears.

To forward network packets between a physical network interface of the TwinCAT/BSD host and a tap(4) device via the `bridge0`, the corresponding devices must become members of the `bridge0`.

The following call makes the physical network interface `igb0` of the TwinCAT/BSD host and the `tap0` device members of the `bridge0` instance:

```
doas ifconfig bridge0 addm igb0 addm tap0
```

Depending on the industrial PC or Ethernet interface used, the naming of the network interface under TwinCAT/BSD may vary and be displayed as `em0`, `em1` or `igb1`, for example.

So that the `bridge0` configuration is already created at system startup, it can be stored in the rc configuration:

```
doas sysrc cloned_interfaces+="bridge0 tap0"
doas sysrc ifconfig_bridge0="addm igb0 addm tap0 up"
```

Via `ifconfig bridge0` the members of the bridge0 can be checked:

```
ifconfig bridge0
```

```
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 58:9c:fc:10:ff:e1
        id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
        root id 00:00:00:00:00:00 priority 32768 ifcost 0 port 0
        member: tap0 flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
                ifmaxaddr 0 port 5 priority 128 path cost 2000000
        member: igb0 flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
                ifmaxaddr 0 port 1 priority 128 path cost 2000000
        groups: bridge
        nd6 options=9<PERFORMNUD,IFDISABLED>
```

The corresponding bhyve(8) command uses only the tap0 instance to connect the virtual machine to the bridge network:

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 2,fbuf,rfb=0.0.0.0:5900,w=1024,h=768 \
-s 3,xhci,tablet \
-s 20,virtio-net,tap0 \
-s 31,lpc \
-A -H -P \
samplevm
```

Filter rules in the Bridged network

By default, the packet filter `pf(8)` under TwinCAT/BSD blocks the exchange of network packets at a `bridge(4)` device.

The filtering behavior on bridge(4) devices can be disabled via sysctl(8). By setting the variables `net.link.bridge.pfil_member` and `net.link.bridge.pfil_bridge` to 0:

```
doas sysctl net.link.bridge.pfil_member=0
doas sysctl net.link.bridge.pfil_bridge=0
```

To set the settings persistently, the following lines must be added to the `/etc/sysctl.conf` file:

```
net.link.bridge.pfil_member=0
net.link.bridge.pfil_bridge=0
```

For more information, see the man pages for bridge(4), sysctl(8), and sysctl.conf(5).

Alternatively, filter rules can be defined for the bridge(4) and its members pf(8) to control packet exchange in the bridge network (see: Firewall [▶ 19]).

NOTE

Connection failure and limited availability in the network

Changes to the filter rules affect the availability of the TwinCAT/BSD host, the virtual machine as well as their services in the network.

9.9.4 Ethernet device passthrough

Industrial PCs with IOMMU support allow physical Ethernet devices to be explicitly assigned to a virtual machine. The virtual machine can thus be connected to a network directly via the physical Ethernet device, without network packets being switched via the TwinCAT/BSD host. The general procedure for assigning PCI devices is described in the chapter PCI device passthrough [▶ 63].

The following call assigns the Ethernet device at the physical PCI address **3:0:0** to the virtual machine PCI address **20**:

```
doas bhyve \
-c sockets=1,cores=2,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd \
-s 0,hostbridge \
-s 20,passthru,3/0/0 \
-s 31,lpc \
-A -H -P -S\
samplevm
```

9.10 PCI device passthrough

Industrial PCs with IOMMU virtualization functions allow physical PCI devices to be explicitly assigned to a virtual machine (see table: Device support for TwinCAT/BSD Hypervisor, device and GPU passthrough. [▶ 49]). PCI devices such as the GPU, network interfaces or USB controllers can be explicitly assigned to a virtual machine as `passthru` devices.

To assign a PCI device to a virtual machine, its PCI address is needed first. The command `pciconf -l` lists all PCI devices and their addresses.

```
$ pciconf -l
...
vgapci0@pci0:0:2:0: class=0x030000 card=0x22128086 chip=0x3e928086 rev=0x00 hdr=0x00
...
igb2@pci0:3:0:0: class=0x020000 card=0x15338086 chip=0x15338086 rev=0x03 hdr=0x00
xhci1@pci0:4:0:0: class=0x0c0330 card=0x00000000 chip=0x8241104c rev=0x02 hdr=0x00
...
```

In the following sample, the three listed devices are to be assigned to a virtual machine.

Device	Description	Address
vgapci0@pci0:0:2:0	GPU	pci0:0:2:0
igb2@pci0:3:0:0	Ethernet Controller	pci0:3:0:0
xhci1@pci0:4:0:0	USB Controller	pci0:4:0:0

To isolate the devices from the TwinCAT/BSD host, they are assigned the `ppt` (**P**CI **P**ass**T**hrough) driver with `devctl`.

```
doas devctl set driver -f pci0:0:2:0 ppt
doas devctl set driver -f pci0:3:0:0 ppt
doas devctl set driver -f pci0:4:0:0 ppt
```

To set the drivers already at system boot the PCI addresses can be added as `pptdevs` to the `/boot/loader.conf` file:

```
pptdevs="0/2/0 3/0/0 4/0/0"
```

A new output from `pciconf -l` now shows that the `ppt` drivers have been assigned to the devices:

```
$ pciconf -l
...
ppt0@pci0:0:2:0: class=0x030000 card=0x22128086 chip=0x3e928086 rev=0x00 hdr=0x00
...
ppt1@pci0:3:0:0: class=0x020000 card=0x15338086 chip=0x15338086 rev=0x03 hdr=0x00
ppt2@pci0:4:0:0: class=0x0c0330 card=0x00000000 chip=0x8241104c rev=0x02 hdr=0x00
...
```

The PCI devices can now be passed to `bhyve` with the parameter `-s [slot],passthru,[slot/bus/function]`. The values for `[slot/bus/function]` refer to the PCI addresses of the `pciconf -l` output. Since passthrough devices use fixed memory addresses, `byhve` must also be passed the flag `-S` as a parameter to disable memory swapping for the process.

The `bhyve` call with assigned on-board GPU, Ethernet and USB controller results for the sample as follows:

```
doas bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd,/vms/samplevm/EFI_VARS.fd \
-s 0:0,hostbridge
-s 2:0,passthru,0/2/0 \
-s 3:0,passthru,4/0/0 \
-s 10:0,virtio-blk,/dev/zvol/vm1_disk0 \
-s 20:0,virtio-net,tap0 \
-s 21:0,passthru,3/0/0 \
-s 31:0,lpc \
-A -H -P -S \
samplevm
```

9.11 Installing a guest operating system using Debian Linux as an example

In the following sample Debian is to be installed in a virtual machine under TwinCAT/BSD. The shell scripts from the GitHub repository https://github.com/Beckhoff/TCBSD_Hypervisor_Samples/tree/main/vm_autostart can be used as a template.

```
vm_autostart
├── Makefile
├── rc.d
│   └── samplevm
└── samplevm
```

The Debian installation requires an advanced virtual machine configuration. The `bhyve` call within the `samplevm` shell script can be adapted accordingly:

```
bhyve \
-c sockets=1,cores=1,threads=1 \
-m 2G \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI.fd,/vms/samplevm/EFI_VARS.fd \
-s 0,hostbridge \
-s 2,fbuf,rfb=0.0.0.0:5900,w=1280,h=1024 \
-s 3,xhci,tablet \
-s 10,nvme,/dev/zvol/zroot/vms/samplevm/disk0 \
-s 15,ahci-cd,/vms/debian-installer.iso,ro \
-s 20,virtio-net,tap0 \
-s 31,lpc \
-A -H -P -w \
"${vm_name}"
```

The bhyve call implies the following configuration steps:

1. the creation of virtual hard disks
2. the use of ISO images
3. booting a UEFI-based virtual machine
4. VNC based interaction with the virtual machine and
5. the configuration of a bridged network

Proceed as follows for the installation:

1. As described in the chapter ZFS data sets as storage location for virtual machines [▶ 56], a ZFS data set is first created for the virtual machine.

   ```
   doas zfs create -p -o mountpoint=/vms/samplevm zroot/vms/samplevm
   ```

2. For the installation of the operating system the Debian "network install" CD-ISO should be used. The ISO file can be downloaded with `fetch(8)`, as described in the chapter Use of installation media (ISO images) [▶ 58], and later passed to the bhyve call as a disk of an ahci-hd device:

```
doas fetch -o /vms/debian-installer.iso https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-11.5.0-amd64-netinst.iso
```

3. To be able to install Debian on a virtual hard disk, the following command creates a ZFS volume which is used as backend for the emulated nvme device in the upper bhyve call:

```
doas zfs create -V 20G zroot/vms/samplevm/disk0
```

4. Debian uses EFI variables to store information about bootable disks. Therefore, a copy of the `/usr/local/share/uefi-firmware/BHYVE_BHF_UEFI_VARS.fd` file should be placed on the ZFS data set for the virtual machine:

```
doas cp /usr/local/share/uefi-firmware/BHYVE_BHF_UEFI_VARS.fd /vms/samplevm/EFI_VARS.fd
```

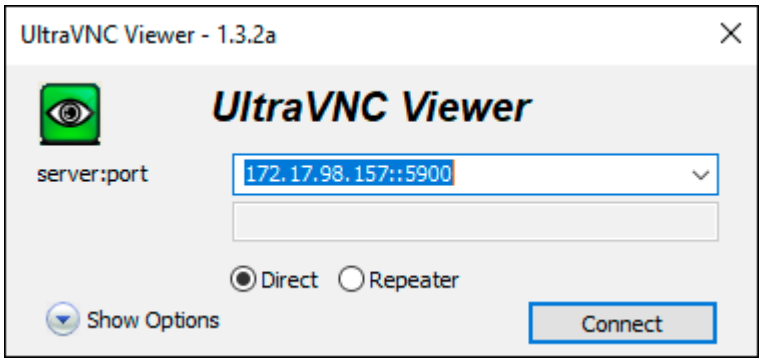
5. For the Debian installation, the virtual machine requires an Internet connection. For this purpose, a bridged network [▶ 60] is created on the TwinCAT/BSD host to which the virtual machine is connected via a virtio-net based network interface and the `tap0` instance.
6. In addition, the virtual machine should be able to be operated via a VNC connection in order to be able to use the graphical installation of the Debian installer at the first start. The samplevm shell script therefore configures the packet filter rules to allow incoming TCP connections on port 5900 of the TwinCAT/BSD host.
7. With the customized `bhyve(8)` call within the samplevm shell script, the virtual machine can be started as follows:

```
doas sh samplevm start
```

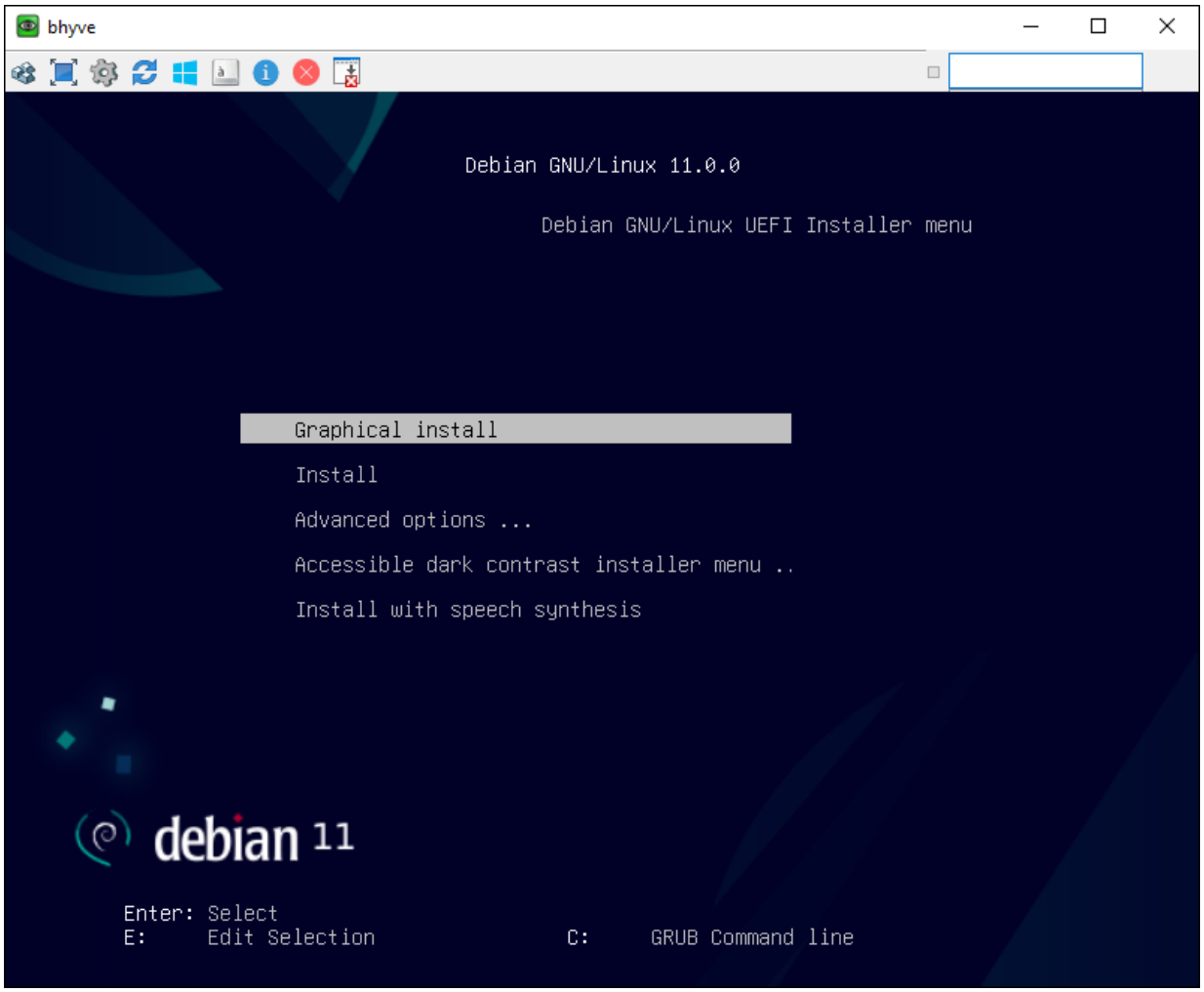
8. The started `bhyve(8)` process then generates the following output on the command line:

```
fbuf frame buffer base: 0x881e00000 [sz 16777216]
```

9. Now a VNC client, such as Ultra-VNC can be used to connect to the virtual machine:



⇨ After successful connection via a VNC client, the installation menu is displayed and the installation of Debian can be started:



Once the installation of Debian is complete, the virtual machine is restarted. This will terminate the VNC connection. After reconnecting, the Debian operating system can be used in the virtual machine.

10 C/C++ projects for TwinCAT/BSD

In this chapter, you will learn how to use executable code

- to compile locally on the Industrial PC under TwinCAT/BSD.

10.1 Compiling under TwinCAT/BSD

This step shows how to generate executable code directly under TwinCAT/BSD with the LLVM compiler. For this purpose, a sample C/C++ project is created, which will use the ADS interface. To use the functions of TcAdsDll in your C/C++ project, you have to integrate the header file TcAdsAPI.h in your project.

The ADS header file is located in the following directory:
usr/local/include/TcAdsAPI.h

Sample C/C++ project `adstest.c`

```
#include <stdio.h>
#include <stdint.h>
#include "TcAdsAPI.h"

int main(){
    printf("ADS Test Sample\n");

    long nTemp;
    AdsVersion* pDLLVersion;
    nTemp = AdsGetDllVersion();
    pDLLVersion = (AdsVersion *)&nTemp;
    printf("Version: %d\n", (int)pDLLVersion->version);
    printf("Revision: %d\n", (int)pDLLVersion->revision);
    printf("Build: %d\n", (int)pDLLVersion->build);
    //printf("Ads DLL Version: %d\n", version);

    long l_port;
    l_port = AdsPortOpen();
    printf("Port opened: %ld\n", l_port);
    AdsPortClose();
    printf("Port closed\n");
    return 0;
}
```

Requirements:

- Install the developer package with the command `doas pkg install os-generic-userland-devtools`

Generate executable code under TwinCAT/BSD **as follows:**

1. Copy the file `adstest.c` to any TwinCAT/BSD directory. Example: */usr/local/ADSinterface*
2. Navigate to the directory with the example file `adstest.c`
3. Use the command `doas cc -c -I /usr/local/include/ -D POSIX adstest.c -o adstest.o` to compile the file `adstest.c`.
4. Link the compiled file with the ADS library with the command `cc -lpthread adstest.o /usr/local/lib/libTcAdsDll.so -o adstest`
5. Execute the file with `./adstest`.

11 TwinCAT 3

11.1 Searching for target systems

Before you can work with the Industrial PC you must connect your local computer to the Industrial PC. After that you can search for the Industrial PC with the aid of the IP address or the host name and subsequently set it as the target system.

To do this the local PC and the Industrial PC must be connected to the same network or directly to each other via an Ethernet cable.

Requirements for this step:

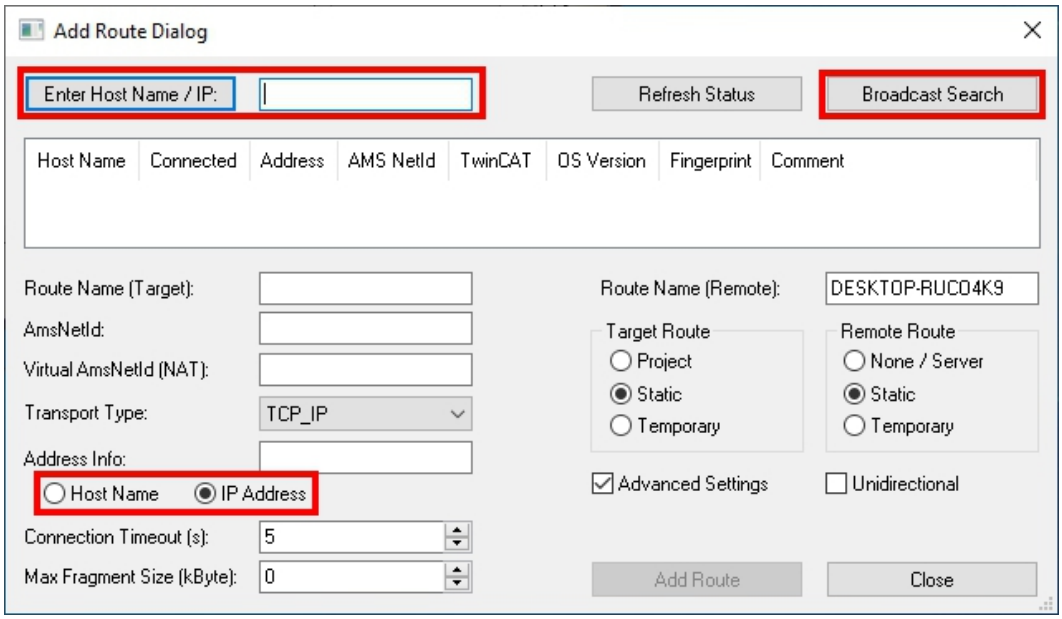
- TwinCAT 3 must be in Config mode.
- The IP address or the host name of the Industrial PC must be known.

Search for a target system as follows:

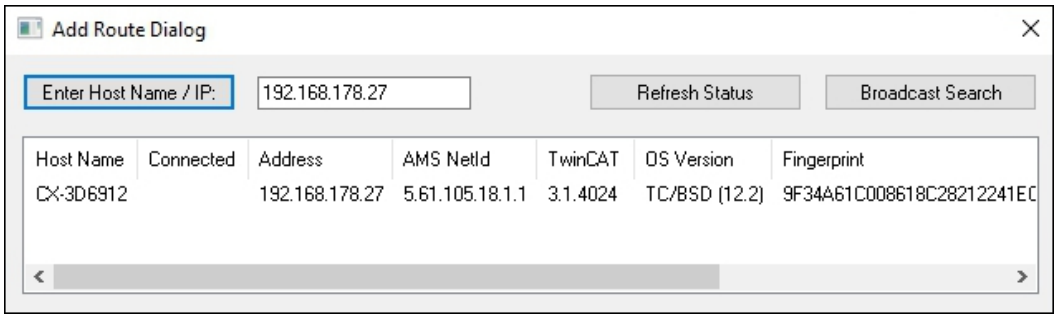
1. In the menu at the top click on **File > New > Project** and create a new TwinCAT XAE project.
2. In the tree view on the left click on **SYSTEM**, and then **Choose Target**.

3. Click on **Search (Ethernet)**.

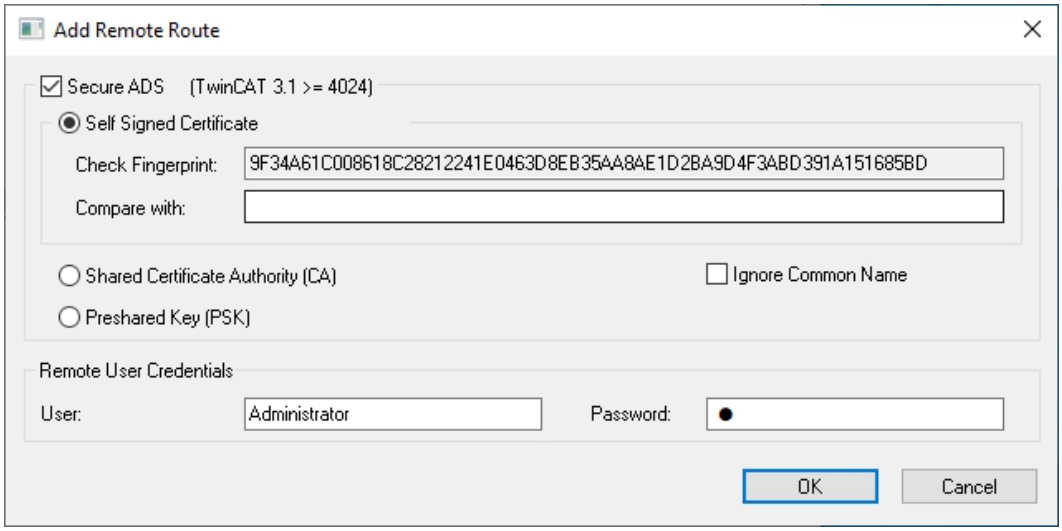
4. Use the **Broadcast Search** function to search for the Industrial PC. The **Broadcast Search** function does not work if the local computer and the Industrial PC are not located in the same subnet.
   In this case, enter the IP address of the Industrial PC under **Enter Host Name / IP** and press **[Enter]**.



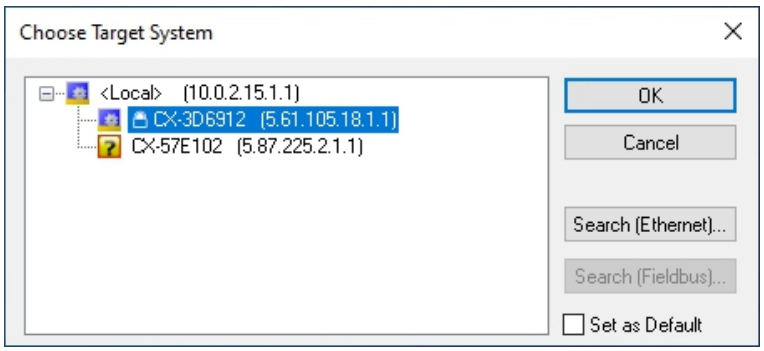
5. Mark the desired Industrial PC and click on **Add Route**.



6. Log in as "Administrator" with the password "1". Note that only Secure ADS is possible by default. For unencrypted ADS connections, you must open the corresponding ADS port in the firewall (see Firewall [▶ 19]).



7. If you do not wish to search for any further devices, click on **Close** to close the Add Route window. The new device is displayed in the **Choose Target System** window.
8. Mark the Industrial PC that you wish to set as the target system and click on **OK**.



⇨ You have successfully searched for an Industrial PC in TwinCAT and inserted it as the target system. The new target system and the host name are displayed in the menu bar.

Using this procedure you can search for all available devices and also switch between the target systems at any time. Next, you can scan the Industrial PC.

11.2 Scan devices

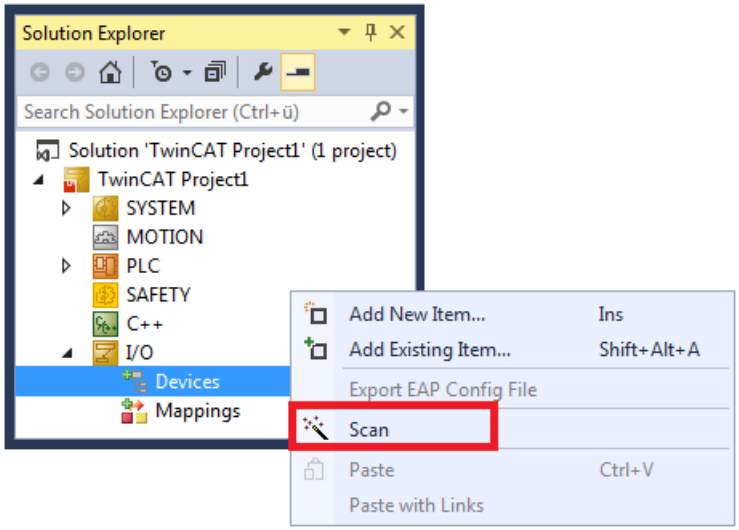
In this work step you will be shown how to scan an Industrial PC in TwinCAT and subsequently configure it.

Requirements for this step:

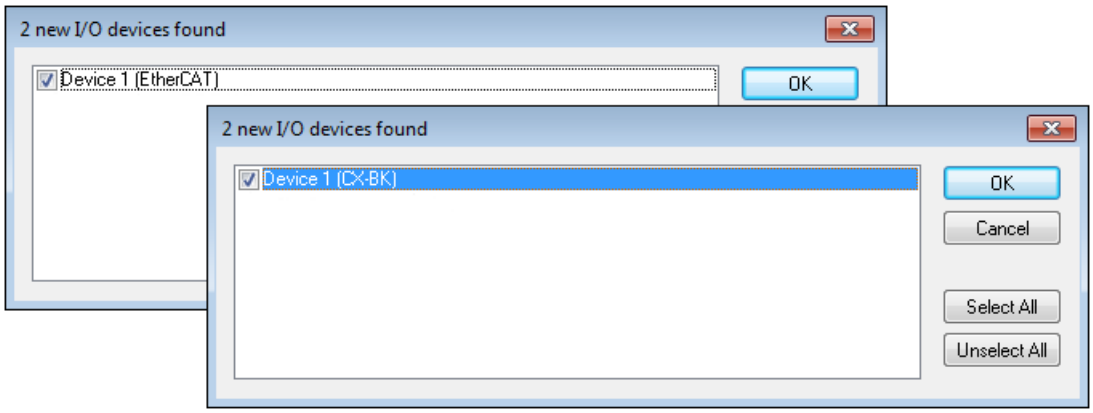
- An Industrial PC has already been selected as the target system.

Add the Embedded PC as follows:

1. Start TwinCAT and open an empty project.
2. In the tree view on the left, right-click on **I/O Devices**.
3. In the context menu click on **Scan**.



4. Select the devices you want to use and confirm the selection with **OK**.
   Only devices that are actual available are offered for selection.



   For Embedded PCs with connected Bus Terminals (K-bus) a Bus Coupler device (CX-BK) is displayed. With EtherCAT terminals (E-bus) the EtherCAT coupler will be displayed.

5. Confirm the request with **Yes**, in order to look for boxes.

6. Confirm the request whether to enable FreeRun with **Yes**.

⇨ The Industrial PC was successfully scanned in TwinCAT and is displayed in the tree view with the inputs and outputs.

11.3 Changing the AMS NetID

This work step shows how you can change the AMS NetID of the Industrial PC. Note that in doing so you will also change the address of the Industrial PC in the TwinCAT network. The AMS NetID consists of 6 bytes and is represented in dot notation and in the hexadecimal system.

Requirements:

- Access rights to the file *TcRegistry.xml*

Proceed as follows:

1. Enter the command `doas ee /usr/local/etc/TwinCAT/3.1/TcRegistry.xml` in the console. The file `TcRegistry.xml` opens.

```
<Value Name="CurrentVersion" Type="SZ">3.1</Value>
<Key Name="System">
      <Value Name="RunAsDevice" Type="DW">1</Value>
      <Value Name="AmsNetId" Type="BIN">053B151A0101</Value>
</Key>
```

2. Change the AMS NetID under the entry `<Value Name="AmsNetId" Type="BIN">053B151A0101</Value>.`
   The entry 053B151A0101 corresponds to the following AMS NetID: 5.59.21.26.1.1
3. Press **[Esc]** and save the changes.

⇨ You have successfully changed the AMS NetID. The new AMS NetID is only applied after a restart of TwinCAT system service. This can be done via the command line with the command `doas service TcSystemService restart`.

11.4 Put TwinCAT into Run or Config mode

You can put TwinCAT to Run or Config mode directly from TwinCAT/BSD, i.e. with the help of the console. Control is provided by the tool `TcSysExe.exe`, which also provides information on licenses, different versions and system IDs. Retrieve more information with `TcSysExe.exe --help`.

Proceed as follows:

1. Enter the command `doas TcSysExe.exe --config` in the console to put TwinCAT into Config mode.
2. Enter the command `doas TcSysExe.exe --run` in the console to put TwinCAT into Run mode.

⇨ The command `TcSysExe.exe --mode` displays the current TwinCAT status in the console.

11.5 Create or delete ADS routes manually

This step describes how you can manually create or delete an ADS route directly from TwinCAT/BSD. To configure ADS routes from TwinCAT/BSD, the tool `ads` can be used. The command `ads` displays all available parameters. Already existing ADS routes are listed in the file `StaticRoutes.xml`.

These settings are alternatively possible via the web interface of the Beckhoff Device Manager (see: Beckhoff Device Manager: web interface [▶ 42]).

Proceed as follows:

1. Create the ADS route according to the following pattern:

```
[<target[:port]>] [OPTIONS...] <command> [CMD_OPTIONS...] [<command_parameter>...]
```

2. At `<target>`, use the host name, IP address, or Ams Net Id of the target system to create a new ADS route.
3. For `<command>` use the command `addroute` and the following options:

```
--addr=<hostname> or IP address of the routes destination
--netid=<AmsNetId> of the routes destination
--password=<password> for the user on the remote TwinCAT system
--username=<user> on the remote TwinCAT system (optional, defaults to Administrator)
--routename=<name> of the new route on the remote TwinCAT system (optional, defaults to --addr)
```

4. Enter the command `ads 192.168.0.231 addroute --addr=192.168.0.1 --netid=192.168.0.1.1.1 --password=1 --routename =example.beckhoff.com` in the console.

⇨ Create new ADS routes according to the pattern shown or delete the unneeded ADS routes in the file `StaticRoutes.xml` under the entries `<Route>`.

```
---snipped---
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <RemoteConnections>
            <Route>
                  <Name>example.beckhoff.com</Name>
                  <Address>192.168.0.1</Address>
                  <NetId>192.168.0.1.1.1</NetId>
                  <Type>TCP_IP</Type>
                  <Flags>64</Flags>
            </Route>
            <Route>
                  <Name>DESKTOP-RUCO4K9</Name>
                  <Address>192.168.40.88</Address>
                  <NetId>192.168.2.15.1.1</NetId>
                  <Type>TCP_IP</Type>
                  <Flags>64</Flags>
            </Route>
      </RemoteConnections>
</TcConfig>
---snipped---
```

11.6 Increase heap memory

NOTE

Oversized heap memory

If the heap memory is too large, the entire main memory is allocated, which leads to the system not working properly. Make sure that the heap memory is not selected too large in relation to the available main memory.

If the heap memory is too small for the download of a PLC project, a corresponding error is issued and the process is aborted.

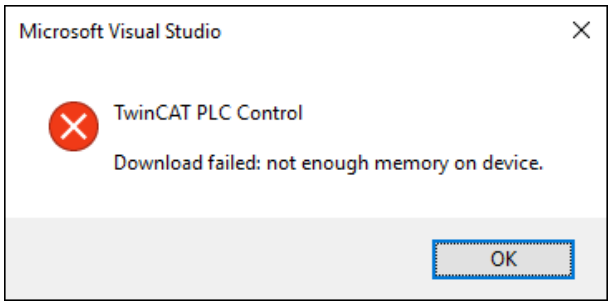


Fig. 9: Error message if the heap memory is too small.

When activating the TwinCAT project where the PLC project is in AutoStart, TwinCAT simply switches back to Config mode and there is no error message or similar.

The size of the heap memory is not automatically adjusted, but can be increased under TwinCAT/BSD for extensive PLC projects in the file `/usr/local/etc/TwinCAT/3.1/TcRegistry.xml`. To do this, the XML file must be expanded as follows:

```
Path: HKEY_LOCAL_MASCHINE\SOFTWARE\BECKHOFF\TWINCAT3\SYSTEM
Entry: <Value Name="HeapMemSizeMB" Type="DW">{size in MB}</Value>
```

Proceed as follows:

1. Enter the command `doas ee /usr/local/etc/TwinCAT/3.1/TcRegistry.xml` in the console. The file TcRegistry.xml is opened.
2. Add the `<Value Name="HeapMemSizeMB" Type="DW">{size in MB}</Value>` entry to the XML file at `<Key Name="System">`.

```
---snipped---
<Key Name="System">
      <Value Name="RunAsDevice" Type="DW">1</Value>
      <Value Name="RTimeMode" Type="DW">0</Value>
      <Value Name="AmsNetId" Type="BIN">0542F70C0101</Value>
      <Value Name="LockedMemSize" Type="DW">33554432</Value>
      <Value Name="SysStartupState" Type="DW">5</Value>
      <Value Name="HeapMemSizeMB" Type="DW">1024</Value>
---snipped---
```

3. The size is set in megabytes. In this sample, these are 1024 MB.

⇨ The settings are applied after a restart with `shutdown -r now` or after restarting the TwinCAT system service with `doas service TcSystemService restart`. After the heap memory is increased to 1024 MB, the PLC project starts and the download is not aborted with an error.

11.7 Adapting the router memory

The main memory is used by TwinCAT/BSD and by TwinCAT (TwinCAT memory). The TwinCAT memory is further divided into the router memory and the PLC memory. The router memory is used for ADS communication and the PLC memory for the actual PLC program including TcConfiguration, mapping and data.

An adjustment of the router memory is only necessary if a large amount of ADS communication takes place and for this reason it becomes necessary to design the size of the router memory accordingly. By default the router memory is set in TwinCAT. The maximum value for the router memory is 1024 MB.

Make sure that the heap memory is larger than the router memory and otherwise increase the heap memory before adjusting the router memory (see: Increase heap memory [▶ 73]). This chapter shows how the router memory can also be customized under TwinCAT/BSD.

To do this, the XML file at `/usr/local/etc/TwinCAT/3.1/TcRegistry.xml` must be adapted as follows:

```
Path: HKEY_LOCAL_MASCHINE\SOFTWARE\BECKHOFF\TWINCAT3\SYSTEM
Entry: <Value Name="LockedMemSize" Type="DW">{size in Byte}</Value>
```

Proceed as follows:

1. Enter the command `doas ee /usr/local/etc/TwinCAT/3.1/TcRegistry.xml` in the console. The file TcRegistry.xml is opened.
2. Adjust the `<Value Name="LockedMemSize" Type="DW">{size in Byte}</Value>` entry in the XML file.

```
---snipped---
<Key Name="System">
      <Value Name="RunAsDevice" Type="DW">1</Value>
      <Value Name="RTimeMode" Type="DW">0</Value>
      <Value Name="AmsNetId" Type="BIN">0542F70C0101</Value>
      <Value Name="LockedMemSize" Type="DW">33554432</Value>
      <Value Name="SysStartupState" Type="DW">5</Value>
      <Value Name="HeapMemSizeMB" Type="DW">1024</Value>
---snipped---
```

3. In this sample a value of 33554432 bytes = 32 MB is set. For example, change the value to 67108864 bytes to increase the router memory to 64 MB.

⇨ The settings are applied after a restart with `shutdown -r now` or after restarting the TwinCAT system service with `doas service TcSystemService restart`.

11.8 Assign isolated cores

For multi-core systems TwinCAT 3 offers the possibility to isolate single cores. This allows different TwinCAT tasks to be assigned to a core isolated for real-time use. This section shows how to set isolated cores in the TwinCAT/BSD console.

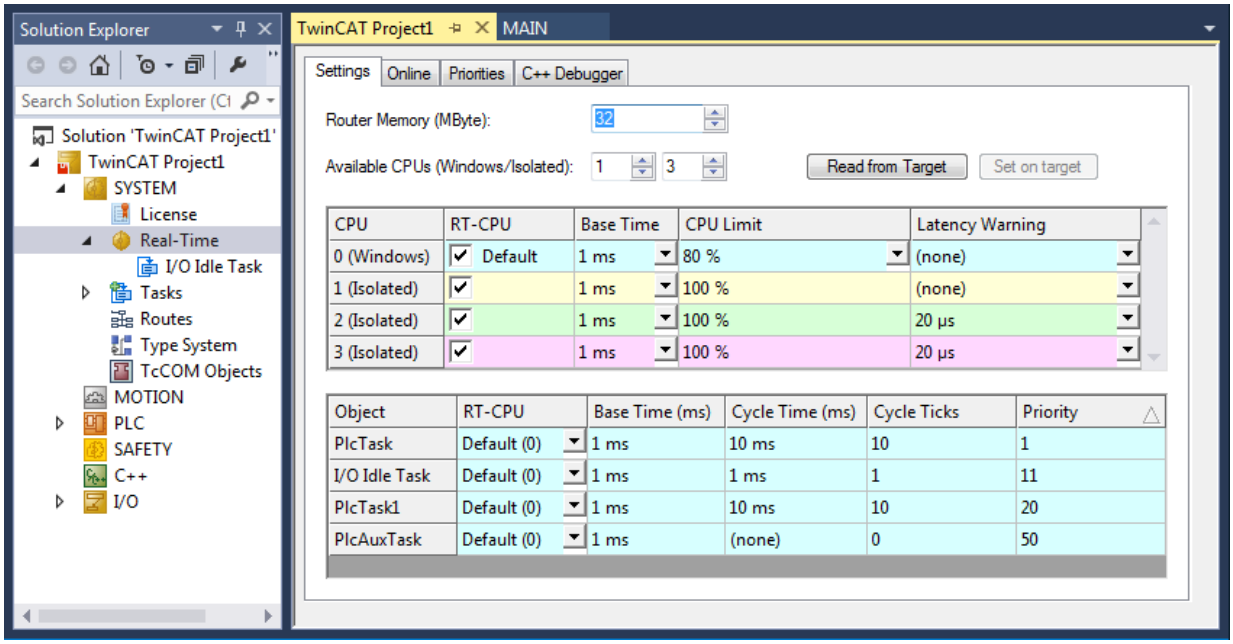
Requirements:

- A multi-core Industrial PC. One CPU core (Shared) for TwinCAT/BSD and three CPU cores (Isolated) should be available for different TwinCAT tasks.

Proceed as follows:

1. With the command `TcCoreConf` an overview of the available cores and their definition Shared/Isolated can be displayed.
2. Enter the command `doas TcCoreConf -s 1` in the console. This sets a CPU core (Shared) for TwinCAT/BSD. The remaining three CPU cores are isolated.
3. Restart the Industrial PC with the command `shutdown -r now` to apply the settings.
4. Then you can use the command `sysctl hw.ncpu` to display the number of CPU cores (Shared).

⇨ You have successfully configured one CPU core (Shared) and three CPU cores (Isolated). The settings can be controlled with the command `TcCoreConf`. You can also read out the current settings from the Industrial PC in TwinCAT 3 (XAE). To do this, click on the button **Read from Target** at **Real-Time**.



With `TcCoreConf --help` all available commands are displayed. With `doas TcCoreConf -d`, for example, all CPU cores can be reset to "Shared".

```
Administrator@CX-3B151A:~ % TcCoreConf -help
TcCoreConf:
   -s --set CPUs
      set number of shared cores
   -d --delete
      set all cores as shared core
   -f NAME, --file NAME
      set name of configuration file to change
   --rsdp ADDR
      set pointer for RSDP
   --show
      show active settings of shared/isolated cores
   --strip
      remove hints of unknown apic-ids
   --noflat
      don't add flat cpu topology setting
```

12 Restore options

Define a backup and recovery strategy for your TwinCAT/BSD system in order to restore TwinCAT/BSD in a very short time in the event of data loss or defective storage media. Backups help to minimize downtime and thus to allow work to continue without large production losses. Both a process for creating a backup copy and a process for restoring it should be defined. Security aspects should also be taken into account and, for example, the storage location where the backup is to be stored should be defined.

Beckhoff offers a simple backup solution with the TwinCAT/BSD installer stick. In addition, the `restorepoint` program makes restore points possible with TwinCAT/BSD; these restore points store the current state of the system and restore it if necessary. A variety of implementations are therefore available, with the exact definition of a backup and restore strategy left to the user.

The following scenarios are possible and are intended to help you understand the different modes of operation. However, the scenarios presented should not be considered the only way recommended by Beckhoff.

Scenario 1: Factory settings

An Industrial PC with TwinCAT/BSD is to be reset to the factory settings in case of a problem.

- The user tests and develops on an Industrial PC with TwinCAT/BSD.
- In the test and development phase, there is a problem because, for example, basic settings have been changed.
- The user solves the problem by resetting TwinCAT/BSD to the factory settings (see:Resetting to factory settings [▶ 77]).

Scenario 2: Series production

The test and development phase has been successfully completed. The machine manufacturer wants to start series production:

- The machine manufacturer creates a restore point (delivery state OEM) in order to be able to restore the system in the event of an error (see: Creating a restore point [▶ 77]. The machine manufacturer's end customer can use this restore point in case of problems.
- The machine manufacturer then activates the Write Filter to secure TwinCAT/BSD in the preconfigured state and to prevent a misconfiguration at the end customer (see: Write filter [▶ 16]).
- In the final step, the machine manufacturer creates a backup, which is stored as a master image and used for series production (see: Creating a backup [▶ 80]).

Scenario 3: Commissioning at the end customer

The machine arrives at the end customer and is to be backed up after commissioning:

- After parameterizing the machine, the end customer creates a restore point called "Commissioning" (see: Creating a restore point [▶ 77]).
- The end customer then activates the Write Filter in order to avoid accidental misconfiguration (see: Write filter [▶ 16]).
- The end customer creates his own backup (see: Creating a backup [▶ 80]) in order to be able to restore the system, for example, in the event of a defective data carrier (see: Restoring a backup [▶ 80]).

12.1 Restore point

Restore points are used to restore an old system state if TwinCAT/BSD exhibits undesirable behavior after a major system change or misconfiguration, and this behavior is not easy to rectify. The advantage of restore points is that these configuration errors are easily and quickly undone without reinstalling TwinCAT/BSD.

You define the time to create a restore point, for example, when you make a larger system change or install third-party programs. However, restore points are not a substitute for a full backup and do not protect against data loss. Regular backups are another protection measure that allows you to protect yourself from data loss due to defective storage media, for example (see: Creating a backup [▶ 80]).

The restore points are created and managed in the console using the `restorepoint` program. The following modes are supported by the program:

- `status`: Lists all available restore points. On delivery, a restore point named `factoryreset`, the Beckhoff factory settings, is available.
- `create`: Creates a new restore point. The name of the restore point can be set as an argument. If no name is specified, an automatically generated name is used.
- `rollback`: Return to a specific restore point. Note that all data created after the restore point will be destroyed. If no restore point is specified as an argument, the user is asked with an interactive dialog.
- `destroy`: The specified restore point is destroyed. In this mode, all existing data is preserved, but the restore point itself is deleted.

Restore points under TwinCAT/BSD are based on ZFS snapshots. As a result, they consume very little memory when they are created. Any change in the saved restore point for the current live system the user is working with is reflected in the memory space used by the restore point. Use `zfs list -t snap` to display all system snapshots.

The USED column shows the actual space used by the snapshot; the REFER column shows the space referenced by the snapshot but actually stored in other datasets. It is therefore always advisable to create a restore point before making any changes in the system, since this hardly uses any system resources. After some time and many changes between the restore point and the live system, it is recommended to delete restore points that are no longer needed in order to free up the increasing memory space used by the restore points.

12.1.1 Resetting to factory settings

You can reset TwinCAT/BSD to the factory settings at any time and restore the delivery status if, for example, the system no longer works properly after a misconfiguration.

The restore points are created and managed in the console using the `restorepoint` program. This section shows you how to reset TwinCAT/BSD to the factory settings.

Proceed as follows:

1. Enter the command `doas restorepoint rollback factoryreset` on the console.
2. All snapshots to which the system is reset are displayed.
3. Confirm the restoration with **[y]**.

⇨ The system is reset to the factory settings. After a restart, TwinCAT/BSD is in the delivery state again.

12.1.2 Creating a restore point

Restore points are used to restore an old system state if TwinCAT/BSD no longer works properly after a major system change or misconfiguration. Create restore points when you want to make major system changes, install programs or run tests.

The restore points are created and managed in the console using the `restorepoint` program. This section shows you how to create restore points in TwinCAT/BSD.

Proceed as follows:

1. Enter the command `doas restorepoint create` on the console.
2. The restore point is created with an automatically generated name.
3. Check the creation of the restore point with the command `restorepoint status` and have all restore points displayed.

```
Administrator@CX-4FAA38$ restorepoint status
last BE: zroot/ROOT/default
factoryreset
2020-08-28T08:56:14Z
2020-08-28T09:03:05Z
```

4. Alternatively, use the command `doas restorepoint create your-restorepoint` in order to define your own name for the restore point.

⇨ The restore point is created and can be used at any time to reset the system (see: Resetting to the restore point [▶ 78]).

```
Administrator@CX-4FAA38$ restorepoint status
last BE: zroot/ROOT/default
factoryreset
2020-08-28T08:56:14Z
2020-08-28T09:03:05Z
your-restorepoint
```

12.1.3 Resetting to the restore point

NOTE

Loss of data

Data and restore points created after a certain restore point are deleted when resetting to a previous restore point.

If TwinCAT/BSD no longer works properly after a misconfiguration, you can easily undo these configuration errors with the help of restore points without reinstalling TwinCAT/BSD.

Proceed as follows:

1. Enter the command `restorepoint status` on the console to display all the restore points that can be used.

```
Administrator@CX-4FAA38$ restorepoint status
last BE: zroot/ROOT/default
factoryreset
2020-08-28T08:56:14Z
2020-08-28T09:03:05Z
your-restorepoint
```

2. Enter the command `doas restorepoint rollback` in the console to see all existing restore points.
3. Select a menu item to reset the system to a specific restore point.

```
Administrator@CX-4FAA38~ $ doas restorepoint rollback
Password:
    1 factoryreset
    2 2020-08-28T08:56:14Z
    3 2020-08-28T09:03:05Z
    4 your-restorepoint
```

4. All snapshots to which the system is reset are displayed.
5. Confirm the restoration with **[y]**.

⇨ TwinCAT/BSD is reset to the restore point and restarted. Note that data and restore points created after the selected restore point are deleted during the reset.

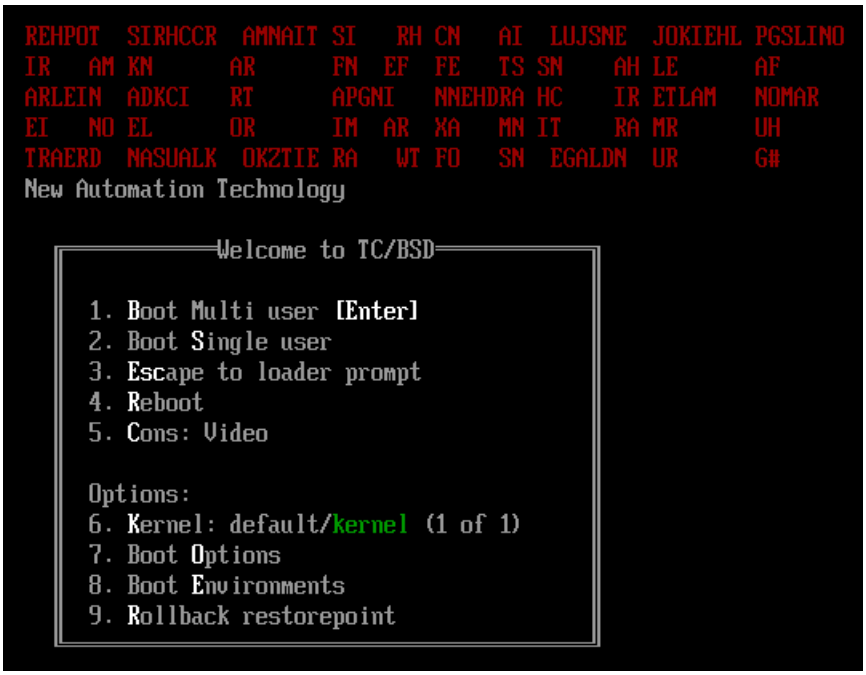
12.1.4 Using the restore boot environment

You can restore a restore point from the restore boot environment when TwinCAT/BSD no longer boots and the console is inaccessible as a result. To do this, start the boot menu during the boot process in order to switch to the restore boot environment.

Proceed as follows:

1. Start the Industrial PC.

2. During the bootup, press and hold the **[Space bar]**. The boot menu appears.



3. Select the option **Rollback restorepoint**.

⇨ TwinCAT/BSD starts in the restore boot environment. Now you can restore the factory settings with the command `restorepoint rollback factoryreset` or use a specially created restore point (see: Resetting to the restore point [▶ 78]).

12.2 Backup and restore

Unlike a restore point, TwinCAT/BSD can be saved and managed as a backup copy on an external storage device by means of a backup.

This backup copy can be used to restore the system in the event of a system failure or data loss. Make regular backups from your system in order to restore your Industrial PC to the state it was at the time of the backup.

12.2.1 Creating a backup

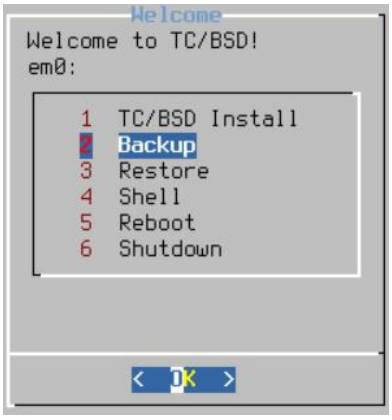
You can create and restore a backup using the TwinCAT/BSD installer stick. All backups are stored on a FAT32 partition on the USB stick. FAT32 is interoperable with Windows and FreeBSD. This allows the backups created to be managed both with a TwinCAT/BSD system and with a Windows system.

Requirements:

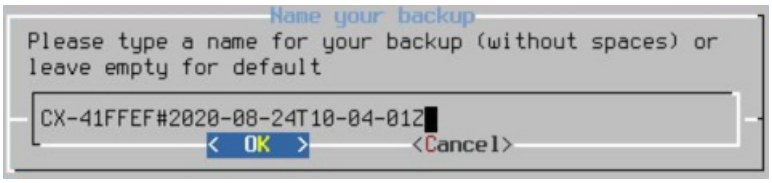
- TwinCAT/BSD installer stick (see: Create bootable USB stick [▶ 9]).

Create a backup as follows:

1. Connect the TwinCAT/BSD installer stick to the Industrial PC.
2. Boot the Industrial PC from the TwinCAT/BSD installer stick.
3. Open the boot menu with **[F7]** if the Industrial PC doesn't boot automatically from the USB stick.
4. Select the UEFI entry for the USB stick and confirm with **[Enter]**. The Industrial PC boots from the USB stick and the Beckhoff TwinCAT/BSD installer is run.
5. Select the option **Backup**.



6. Assign a file name to the backup or accept the default name made up of host name and timestamp.



7. Select the option **Reboot** for a reboot once the backup is complete.

⇨ The backups are stored on the USB stick with the respective file name. Archive the backups on the USB stick. You can also copy the backups to an external storage medium or archive them on the network.

12.2.2 Restoring a backup

You can restore a backup with the aid of the TwinCAT/BSD installer stick. To do this, the Industrial PC must be booted from the TwinCAT/BSDinstaller stick.

Requirements:

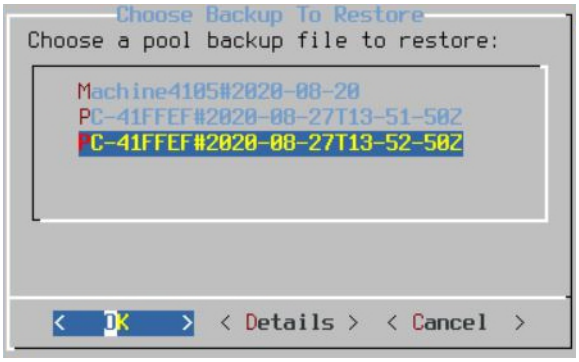
- TwinCAT/BSD installer stick (see: Create bootable USB stick [▶ 9]).

Proceed as follows:

1. Connect the TwinCAT/BSD installer stick to the Industrial PC.
2. Boot the Industrial PC from the TwinCAT/BSD installer stick.

Open the boot menu with [F7] if the Industrial PC doesn't boot automatically from the USB stick.

3. Select the UEFI entry for the USB stick and confirm with **[Enter]**. The Industrial PC boots from the USB stick and the Beckhoff TwinCAT/BSD installer is run.
   Select the option **Restore**.
4. Select the backup to be restored to the Industrial PC.



⇨ Restart the Industrial PC after restoration. The Industrial PC is now in the state it was at the time of the backup.

12.2.3 Creating and restoring a backup from the live system

If required by your application, backups can also be created and restored from the live system, without a TwinCAT/BSD installer stick. Use the scripts TcBackup and TcRestore for this purpose.

Do not create a backup from the running system if the system is writing to the disk at the time of the backup. A backup can become corrupted if the system has write access to the disk during the backup. In other words, make sure that there are no processes running that persistently back up data and that the disk you want to restore your backup to has enough space.

Running TcBackup and TcRestore and writing to and from the file where the backup is saved must be done with root rights. In other words, execute a shell with root rights beforehand, in which you then work, or execute the single command as a string with a shell with root rights. The latter option is illustrated in the following examples.

Proceed as follows:

1. Type the command `doas sh -c "TcBackup.sh --disk /dev/ada0 > backup.bckp"` to create a backup from the disk ada0 to the file Backup.bckp.
2. Enter the command `doas sh -c "TcRestore.sh --disk /dev/ada1 < backup.bckp"` to restore a backup from the file Backup.bckp file on the disk ada1.

⇨ The two commands can be combined, as follows. The command `doas sh -c "TcBackup.sh --disk /dev/ada0 | TcRestore.sh --disk /dev/ada1"` creates a backup of the disk ada0 and immediately restores it to the disk ada1.

13 Error handling and diagnostics

13.1 Using kernel messages for diagnosis

You can turn on kernel messages for your own diagnostic purposes or for Beckhoff Support. The kernel messages are displayed when booting from TwinCAT/BSD and stop at the error location. As a result, you or Beckhoff Support can locate errors.

Requirements:

- Restart required. The kernel messages can only ever be turned on during the bootup.

Proceed as follows:

1. Start the Industrial PC.
2. Press the **[Spacebar]** during the boot process.
   The boot process is paused.
3. Press button **[6]** to select option **6. Boot Options**.
   The **Boot Options** menu appears.
4. Press button **[5]** to activate option **5. Verbose: on**.
5. Press **[Enter]** to conclude the procedure.

⇨ TwinCAT/BSD continues to boot up and the kernel messages are displayed. These settings are not saved. Repeat the work steps shown if you require the kernel messages again at the next start.

13.2 Log files

Kernel, security and TwinCAT logs can be used for diagnostic purposes and are located in the following directories.

General kernel log
/var/log/messages

Open the kernel log with the following command:

```
cat /var/log/messages
```

Security log
/var/log/security

Open the security log with the following command:

```
doas cat /var/log/security
```

Filter the output of `cat` with the pipe operator and the program `grep`. The pipe operator passes the output of `cat` to the program `grep`, which can filter its input based on an expression. Use the expression `cat /var/log/messages | grep -i tc` to filter for all TwinCAT-relevant data. The option `-i` is case-insensitive.

The program `less` can be used to output and search text files. For example, **[&]** can be used to start the search function. Use **[h]** to display information on further functions. Use **[q]** to quit the program.

Another useful program pertaining to log files is `tail`. In this case only the last few entries are output. This is useful if only the most recent log entries are of interest. For example, the command `tail -5 /var/log/messages` outputs the last five entries of the log file.

13.3 Dumps

Kernel dump

The kernel dump can be found under
/var/crash

13.3.1 Using automatic process dump

When a process crashes, a memory dump is automatically created and stored in the file `progname.core`. The file contains the full process state at the time of the crash. The file is usually stored where the user-mode process is located: */usr/local/bin*

The file can then be analyzed with a debugger such as `gdb`. The following section describes the local analysis of the dump directly on the TwinCAT/BSD system. Of course, the analysis can also be performed with suitable programs on the Windows development computer. To do this, copy the dump to your Windows computer using WinSCP, for example.

Requirements:

- Install the debugger `gdb` with the command `pkg install gdb`.

Proceed as follows:

1. Navigate to the appropriate directory if you are not in the same directory as the file containing the process dump.
2. Enter the command `gdb -c <filename>` on the console to examine a process dump.

⇨ Enter `help` to see more information about the name of the GDB command or general information about GDB.

13.3.2 Creating a process dump manually

rUse the `gcore` program to examine failure-prone programs, or if your Industrial PC is in an infinite loop or something situations. The process dump is particularly useful for taking a snapshot of a running process and analyzing processes under TwinCAT/BSD.

By default, the process dump is written to the file `core.pid`. The file can then be analyzed with a debugger such as `gdb`.

Proceed as follows:

1. Determine the process ID (`pid`) of the desired process with the command `pgrep -l <processname>`. The command `ps -A` can be used to list all processes.
2. Enter the command `gcore <pid>` in the console. Sample: `gcore 6674`
3. The command `gcore 6674` generates a file with the name `core.6674`

⇨ The file is created in the current directory. This file can then be read and analyzed with a debugger. The option `-c` can be used to specify your own file name.
Sample: `gcore -c testfile 6674`

13.4 Using the ADS monitor

The TwinCAT ADS monitor is divided into two applications. The AMS logger records the AMS commands, while the AMS viewer displays the recorded data or can be used to control the AMS logger remotely.

The AMS logger is called up and configured with the program `tcamslog`. Additional parameters can be used determine the maximum size of the log file or whether a ring buffer should be used, for example.

Table 4: ADS monitor, parameters of the `tcamslog` application.

Parameter	Description
`-l`	`listen` Waits for an AMS viewer connection.
`-p`	`port` Port for the AMS viewer connection. Standard: 0xbf12/48914
`-c`	`capture` Starts logging AMS commands.
`-f`	`file` Name of the log file. Standard: `ams.cap`
`-d`	`dir` Directory in which the log file is stored. The default is the current directory.
`s-`	`size` Maximum size of the log file. Standard: 15 MB
`r-`	`ringbuffer` Enabled by default. The log is distributed over two or more files.

Proceed as follows:

1. Enter the command `tcamslog -c -r -s 20 -f testlog` in the console to record AMS commands.
2. The parameters used in the sample determine the size `[-s]`, the file name `[-f]` and the use of the ring buffer `[-r]`.

⇨ In the next step, load the recorded AMS commands into the AMS viewer to analyze the log file.

13.5 Analyzing network traffic with Wireshark

TwinCAT/BSD has a packet sniffer as standard. The `tcpdump` program monitors the Ethernet interfaces, records the network traffic and saves the data in a file on the Industrial PC.

The saved file can then be copied to a development computer, opened with Wireshark and analyzed.

Requirements:

- Wireshark installed on the development computer: https://www.wireshark.org/download.html
  Wireshark user guide: https://www.wireshark.org/docs/wsug_html_chunked/

Proceed as follows:

1. Enter the command `doas tcpdump -i igb1 -s 0 -w DHCP.dump` in the console. In this example, igb1 corresponds to the Ethernet interface X000.
   -i Ethernet interface.
   -s length of the snapshot. The value "0" sets the length to the default value of 262144 bytes.
   -w file in which the output should be stored.
2. Confirm the command with the administrator password.

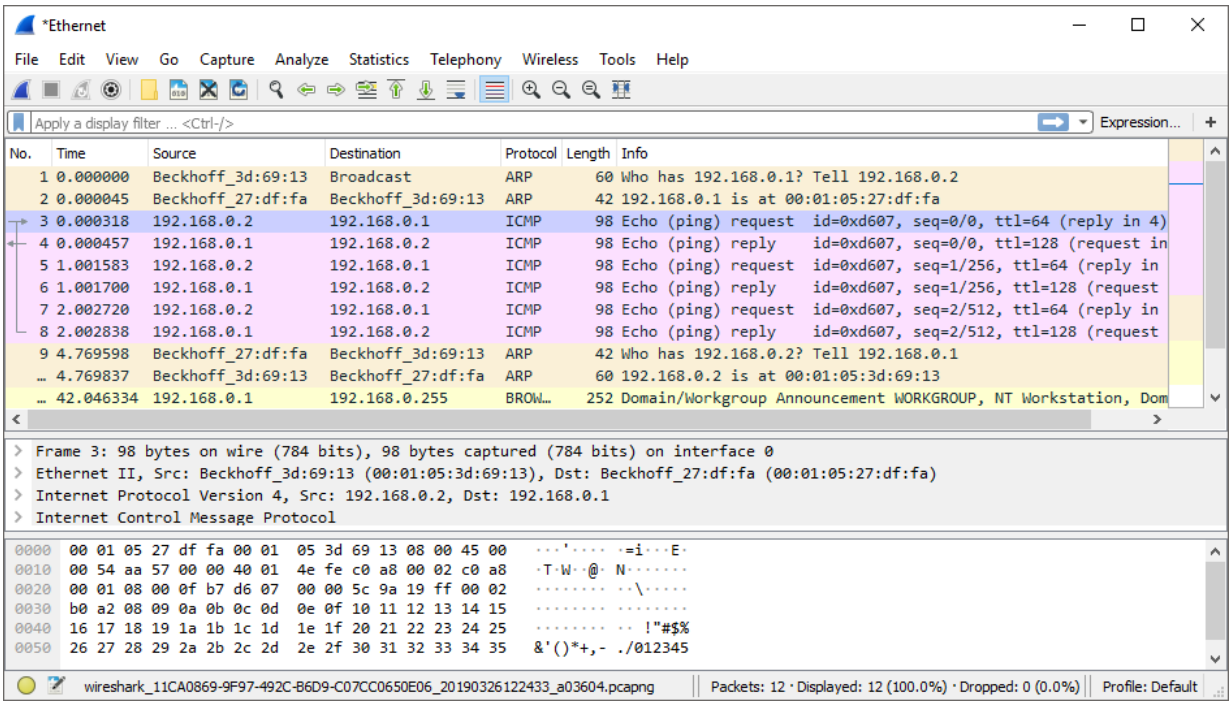
```
tcpdump: listening on igb1, link-type EN10MB (Ethernet), capture size 262144 bytes
```

3. You can stop recording at any time by pressing **[Ctrl] + [c]**.

```
33523 packets captured
33531 packets received by filter
0 packets dropped by kernel
Administrator@CX-3B151A:~ %
```

4. In this example, the file **DHCP.dump** is saved in the home directory under */usr/home/Administrator*.
5. Copy the file **DHCP.dump** to a development computer using the WinSCP client (see: Managing files with the WinSCP client [▶ 45]).

⇨ You have successfully recorded the network traffic on the Ethernet interface igb1. You can then open and analyze the file **DHCP.dump** with Whireshark.



13.6 System repair

If TwinCAT/BSD does not boot due to an inconsistent file system or error in a configuration file, there are two ways to repair the TwinCAT/BSD installation or recover data.

For repair or data recovery, use either the

- TwinCAT/BSD installer stick you used for the installation
- or single-user mode, which you can run during system startup.

13.6.1 Booting from the USB installer stick

NOTE

Security risk

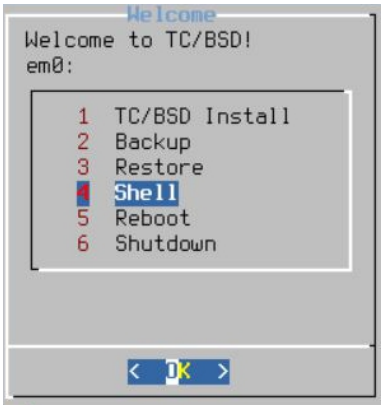
In the default setting, every user who has physical access to the Industrial PC has root rights and thus full control over the system. Restrict access to the Industrial PC to trusted persons.

The TwinCAT/BSD installer stick can be used for repair or data recovery, for example if a faulty process prevents a system start or a faulty TwinCAT project causes a boot loop.

When you boot from the TwinCAT/BSD installer stick, you have access to a fully functional TwinCAT/BSD system installed on the USB stick. In contrast to single-user mode, this allows you to copy important data directly to the USB stick or create a backup after a repair.

Proceed as follows:

1. Boot from the TwinCAT/BSD installer stick and select the option **shell**.



2. Import the memory pool (zpool) of your TwinCAT/BSD system with the command `zpool import -fR /mnt zroot`.
3. First, mount the default dataset with the command `zfs mount zroot/ROOT/default`. All other datasets of your TwinCAT/BSD system can be mounted with the command `mount -a` or individually, specifying the dataset.

⇨ The file systems of the broken system can now be accessed via `/mnt`.

13.6.2 Start single-user mode

NOTE

Security risk

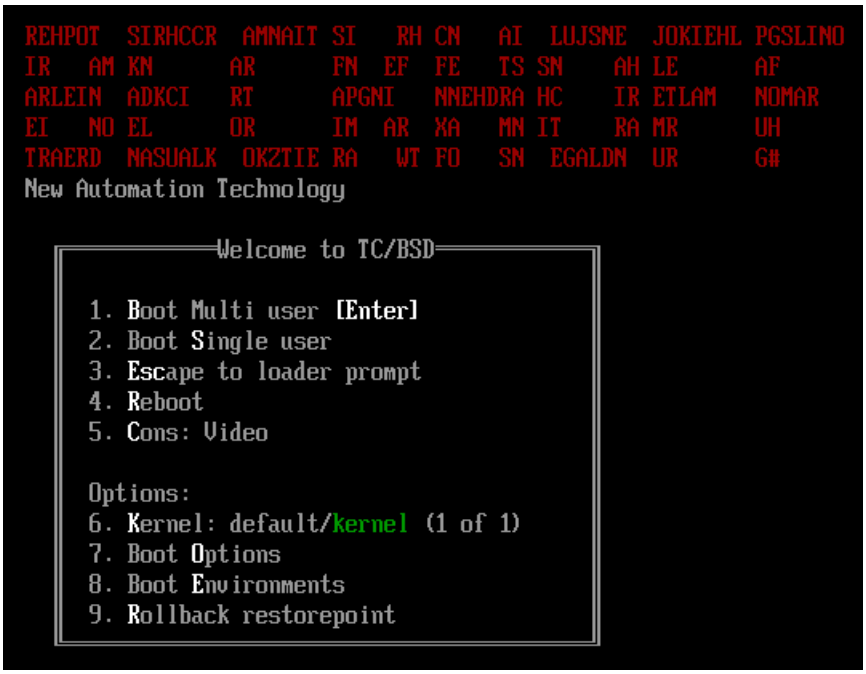
In the default setting, every user who has physical access to the Industrial PC has root rights and thus full control over the system. Restrict access to the Industrial PC to trusted persons.

Single-user mode grants full access to the local system and the configuration files and can be executed during system startup. Single-user mode is similar to safe mode under Windows and can be used for repair or data recovery, for example if a faulty process prevents a system start or a faulty TwinCAT project causes a boot loop.

In single-user mode you have no access to the network and no processes are running. Processes and the network can be started if required.

Proceed as follows:

1. During the bootup, press and hold the **[Space bar]**. The boot menu appears.
2. Select single-user mode with **[2]**.



3. The dataset `zroot/ROOT/default` is mounted automatically. All datasets can be mounted collectively with the command `zfs mount -a`, or the datasets can be mounted individually.
4. Enable the writing rights for the dataset `zroot/ROOT/default` with the command `zfs set readonly=off zroot/ROOT/default` in order to be able to make changes to the system.

⇨ You can exit single-user mode again with `exit`.

14 Appendix

14.1 Important commands

This chapter summarizes and explains important and frequently used commands. The summary is intended as an aid and does not claim to be exhaustive.

14.1.1 TwinCAT

Table 5: Important commands and tools, TwinCAT.

Command	Description
`ads`	The tool can be used to manage ADS routes. In addition, variables can be read, written or license information, such as the system ID) can be read. Enter the command `ads` in the console to view all parameters related to the tool. Sample Create ADS route: `ads 192.168.0.231 addroute --addr=192.168.0.1 --netid=192.168.0.1.1.1 --password=1`
`TcSysExe.exe`	With TcSysExe.exe it is for example possible to control the TwinCAT mode from the console and to put TwinCAT into Run or Config mode. In addition, information on licenses, different versions and system IDs can be retrieved. The command `TcSysExe.exe --help` displays all available parameters.
`TcRegistry.xml`	In the file `TcRegistry.xml` the Ams Net Id, the HeapMemSize and LockedMemSize can be set.
`StaticRoutes.xml`	The file `StaticRoutes.xml` is used to configure ADS routes from TwinCAT/BSD.
`TcCoreConf`	With the tool it is possible to manage CPU cores and isolate CPU cores, for example. Enter the command `TcCoreConf --help` in the console to view all parameters.
`TcRteConfig`	This tool can be used to disable real-time Ethernet when you do not need real-time communication. Enter the command `TcRteConfig --help` in the console to view all parameters.

14.1.2 Shell

Table 6: Important commands, shell in general.

Command	Description
`script`	Creates a typescript of the terminal session.
`which <command>`	Search for the command `<command>` in the current directory and display where it was found.
`history 20`	Display the last 20 commands that were entered.
`!<num>`	Execute the command `<num>` again from the history.
`<command1>; <command2>`	Execute command 1 followed by command 2.
`<command1> && <command2>`	Execute `<command1>`, followed by `<command2>`, but only if `<command1>` was successful ($? = 0).
`<command1> | <command2>`	Redirect output of `<command1>` to input of `<command2>`.
`<command> >&out.txt`	Send both the standard output and the error output of a command to the file `out.txt`.
`printenv`	Display all environment variables.
`echo $PATH`	Display individual environment variable "PATH".
`setenv <variable> "value" [csh]`	Sets environment variable <variable>.
`unsetenv <variable> [csh]`	Removes environment variable <variable>.
`^C (Ctrl-C)`	Terminate current command.

Command	Description
^U (Ctrl-U)	Delete up to the beginning of the line.
reset	Reset terminal settings.
exit logout	Exit shell.

Table 7: Important commands, job control.

Command	Description
^C (Ctrl-C)	Terminate current foreground process.
^Z (Ctrl-Z)	Suspend current foreground process. Creates a suspended job.
jobs	List jobs under this shell.
kill %<num>	Terminate the job with the number <num>.
fg fg %<num>	Restart suspended process in the foreground.
bg bg %<num>	Restart suspended process in the background.
<command> &	Start the command as a background job.

14.1.3 File and directory management

Table 8: Important commands, file management.

Command	Description
`less <file>`	The file content is read out Space bar = next page, b = previous page, q = exit / = forward search, ? = backward search, n = repeat search
`grep -i <string> <file>`	Shows all lines containing the specified string; **-i=** is case sensitive.
`wc -l <file>`	Counts the lines in the file.
`tail -f <file>`	Particularly useful for log files. The last 10 lines of the file are displayed. The parameter -f also displays newly added lines. Exit with ^C.
`Tail -n <file>`	This can be used to adjust the number of lines that are output. Example: tail -n <file> shows only the last line of the file.
`strings <file> | less`	Extracts strings from a binary file.
`touch <file>`	Creates a file if not already present, or updates the timestamp.
`rm <file>`	Delete file.
`cp <file> <user>`	Copy file.
`cp <file1> <file2> ... <path>/`	Copy one or more files to another directory. The trailing slash after <path> is not essential, although it prevents errors when copying a file if the path does not exist.
`mv <oldname> <user>`	Rename file or directory.
`mv <file1> <file2> ... <path>/`	Move one or more files to a directory.
`ln <file> <user>`	Create a hard link from <file> to <user> (both names point to the same inode of the file system). Both names must be on the same file system.
`ln -s <path> <user>`	Make <user> a symbolic or soft link that points to the path that can be a file or directory and can be anywhere in the file system.

Table 9: Important commands, file permissions.

Command	Description
`ls -l <file>`	Displays permissions for files or directories. -rwxrwxrwx For a file: **r** allows reading; **w** allows writing/attaching; **x** allows executing. For a directory: **r** allows listing of content; **w** allows creating or deleting files within the directory; **x** allows entering the directory.
`ls -ld <path>`	Directories are displayed like files. Without -d, the directory content is listed recursively when directories are entered.

`chown <user> <path>`
`chgrp <group> <path>`
`chown <user>:<group> <path>`
	Changing the owner, group or both of a file or directory.

`chmod [ugoa]+[rwx] <path>` `chmod [ugoa]-[rwx] <path>`	Adding or removing permissions. **u** = user (owner), **g** = group, **o** = others, **a** = all (ugo) e.g. "chmod go+r file" adds the permission 'r' to 'group' and 'others'.
`chmod <nnn> <path>`	Change all bits simultaneously to the octal value nnn. e.g. "chmod 640 file" sets rw- for user, r-- for group, --- for others. 0 --- 1 --x 2 -w- 3 -wx 4 r-- 5 r-x 6 rw- 7 rwx
`umask` `umask <nnn>`	Show or set the file creation mask for this session; these are the permission bits that are not set for newly created files. For example, “umask 022” means that newly created files have no more than rwxr-xr-x permissions.

Table 10: Important commands, file search.

Command	Description
`find <path> -type f`	Finds all files under the specified path. Use "." for the current directory. Use the option `-type f` to only display files.
`find <path> -type f -name 'placeholder*'`	Finds all files under the specified path whose name begins with "placeholder.
`find <path> -type f | xargs <command>`	Find all files under the path and apply `<command>` to each of them.
`find <path> -type f -print0 |` `xargs -0 <command>`	Secure version of the above command, and works with file names containing spaces.

Table 11: Important commands, compressed files and archives.

Command	Description
`gzip -dc <file>.gz | less` `bzip2 -dc <file>.bz2 | less`	Reads a compressed text file without unpacking it on the hard disk.
`tar -tzf <file>.tgz or .tar.gz` `tar -tjf <file>.tbz2 or .tar.bz2`	Shows the contents of the compressed tar archive. Add option `-v` for more details.
`tar -xvzf -C <path> <file>.tgz` `tar -xvjf -C <path> <file>.tbz2`	Extract the contents of the compressed archive to the specified directory, otherwise to the current directory.

Table 12: Important commands, directories.

Command	Description
`pwd`	Display current directory.
`cd <path>`	Change to a subdirectory of the current directory.
`cd ..`	Move up one level to the parent directory.

`cd /`
`cd /<absolute path>`
`cd ~<user>`
`cd`
	Change current directory: to the root directory, to an absolute path, to the home directory of a particular user, or to your own home directory.

`ls` `ls <path>`	Lists the contents of the current directory or the specified directory.
`ls -l`	Lists the directory in long form.
`ls -a`	Lists all files, including hidden files.
`ls -d`	Lists the directory itself, instead of its contents.
`ls -ld <path>`	Sample for the combination of flags.
`mkdir <path>`	Create a directory.
`rmdir <path>`	Delete an empty directory.
`rm -rf <path>`	Recursively delete a directory and its entire contents.

14.1.4 System administration

Table 13: Important commands, user accounts.

Command	Description
`id`	Show current `uid`, `gid` and additional groups.
`whoami`	Display current user name only.
`cat /etc/passwd`	Show all user accounts.
`cat /etc/group`	Show all groups.
`pw useradd <user> -m`	Create user; **-m**= create home directory.
`passwd` `passwd <user>`	Set or change password for yourself or another account (administrator only).
`pw usermod <user> -G wheel`	Add users to the group "wheel" or simply edit /etc/group directly.
`pw userdel <user> -r`	Delete user; **-r**= remove home directory and all content.
`cat /etc/master.passwd`	View all accounts, including encrypted passwords.
`vipw`	Lock master.passwd Edit it and rebuild password databases.

Table 14: Important commands, file system.

Command	Description
`mount`	Display mounted file systems.
`df` `df -h`	Shows occupied and free space in all mounted file systems. Adding **-h** = shows 1G instead of 1048576.
`du -c <path>`	Adds space occupied by files or directories in the specified path or in the current directory.
`mount -r -t cd9660 /dev/acd0 /cdrom`	Mount device */dev/acd0* [IDE CD] in directory */cdrom*; file system type is cd9660;**-r**=read-only.
`umount /cdrom`	Eject the device. The device must not be used.
`fstat`	List processes with open files.
`cat /etc/fstab`	Display file system table.
`mount /cdrom`	Mount */cdrom* with parameters from */etc/fstab*
`mount -a`	Mount all file systems in /etc/fstab except those marked "noauto" (this happens during normal booting, but is useful when booting in single-user mode).

Table 15: Important commands, packages.

Command	Description
`pkg info`	Display an overview list of the installed packages.
`pkg info <package>`	Display a detailed description of the package.
`pkg info -l <package>\*`	Display a list of all files contained in the package.
`pkg add <file>-1.2.3.tbz`	Install package from file.
`pkg add -r <package>`	Install package from the default FTP server.
`PACKAGEROOT="ftp://ftp.uk.freebsd. org" pkg add -r <package>`	Install package from an alternative FTP server.
`pkg install <package>`	Installs package from remote repository or local archive.
`pkg delete <package>`	Uninstalling a package.

Table 16: Important commands, kernel modules.

Command	Description
`kldstat`	Display loaded modules.
`kldload <module>`	Load the named module and all modules on which it depends.
`kldunload <module>`	Unload module.

Table 17: Important commands, network.

Command	Description
`ifconfig`	Display all interfaces.
`ifconfig igb0 192.168.0.1/24`	Configure interface.
`netstat -r -n`	Display table with redirects.
`route add default 192.168.0.254`	Add a static default route.
`ping <IP-Adress>`	Send test packages. Terminate with ^C.
`traceroute -n <IP-Adress>`	Send test packets and display intermediate routers.
`tcpdump -i igb0 -n -s1500 -X` `tcpdump -i igb0 -n tcp port 80 -w <file>`	Displays complete packets that were sent and received via a specific interface. The second form shows only package headers to/from TCP port 80. Use the option -w <file> to save the network dump in file <file>.
`/etc/rc.d/netif start`	Initialize network interfaces using the settings in /etc/rc.conf.
`/etc/rc.d/routing start`	Initialize static routes from the settings in /etc/rc.conf.
`/etc/rc.d/dhclient start`	Configure interfaces marked with "DHCP" in /etc/rc.conf.
`netstat -finet -n`	Display active network connections. Use -a to add listening sockets.
`sockstat -4 -l`	Displays processes that listen to IPv4 and IPv6 sockets.

Table 18: Important commands, processes.

Command	Description
`ps aux`	Show all processes.
`ps aux | grep <process-name>`	Show all processes that correspond to the pattern <processname>. Note that grep <processname> can be displayed itself.
`top`	Continuous display of the most active processes. Quit with q.
`kill <pid>`	The process with the specified process ID is quickly cleaned up and terminated.

Table 19: Important commands, system status.

Command	Description
`Alt-F1 ... Alt-F8`	Switch between virtual consoles.
`date`	Display current date and time.
`ntpdate -b <server1> <server1> ...`	Synchronize the clock with the specified NTP servers.
`uptime`	Display time since last restart and average load.
`w`	Shows who is currently logged in.
`last -10`	Display the last 10 logins.
`shutdown -r now`	Restart.
`doas shutdown -p now`	Switch-off.

14.1.5 Important files and directories.

Table 20: Important files and directories.

Path	Description
/boot/kernel/kernel /boot/kernel/	The kernel itself.

/boot/loader.conf	Kernel modules at startup. See /boot/defaults/loader.conf
	hint.acpi.0.disabled=1 # disable ACPI
	if_wi_load="YES" # load the 'wi' network driver
	snd_driver_load="YES" # load all sound drivers

/dev/null	The "bit bucket". To discard all output of a command (stdout and stderr): # somecommand >/dev/null 2>&1 [sh]
/etc/crontab	Regular scheduled tasks.
/etc/group	Binds additional groups to users (only becomes effective after the next login).
/etc/hosts	Local assignments between IP addresses and host names.
/etc/inetd.conf	Controls services that were started by inet but do not have their own daemon processes, e.g. ftpd.
/etc/localtime	Binary file, not editable. Describes the current time zone. # cp /usr/share/zoneinfo/Africa/Maputo /etc/localtime
/etc/mail/ mailer.conf	Configures which MTA is used when local processes generate emails.
/etc/make.conf	Default values for creating software applications/ports. Only available if created by user. If not present, the default values of the ports are used.
/etc/motd	"Message of the day" is displayed during login.
/etc/newsyslog.conf	Configures the automatic rotation of log files.
/etc/periodic/...	Various scripts that are executed at scheduled times.

/etc/rc.conf	Master configuration file. See /etc/defaults/rc.conf for permitted settings.
	# network settings
	hostname="foo.example.com"
	ifconfig_igb0="192.168.0.1/24" # oder "DHCP"
	defaultrouter="192.168.0.254".
	# set clock on startup
	ntpdate_enable="YES".
	ntpdate_flags="-b ntp-1.example.net ntp-2.example.net".
	# activate services
	inetd_enable="YES"
	sshd_enable="YES".

/etc/rc.d/...	Startup scripts. Run as /etc/rc.d/<script> start or /etc/rc.d/<script> stop Only works if the corresponding service exists. service_enable="YES" in /etc/rc.conf
/etc/rc.local	Create this script to run additional commands at system startup.
/etc/resolv.conf	Configuring the DNS client example.com nameserver 192.0.2.1 nameserver 192.0.2.2.2
/etc/ssh/ sshd_config	Configure ssh daemon to allow or deny root logins, for example.
/etc/sysctl.conf	Sets runtime kernel variables at startup: net.inet.ip.forwarding=1 # if this system is a router.
/etc/syslog.conf	Configure the destinations of log messages. After the change:

Path	Description
	`# killall -1 syslogd`
`/etc/ttys`	Configure logins on serial lines or modems.
`/rescue/...`	Statically linked binary files for use in emergencies.
`/root`	Home directory for 'root' users (still available when other file systems are not mounted).
`/usr/local/etc/...`	Configuration files for third-party programs (ports/packages).
`/usr/share/skel/...`	Placeholders that fill the home directory of a new user.
`/var/db/pkg/...`	Path under which the installed packages `pkg` are stored (do not change this!).
`/var/log/maillog`	Mail log file.
`/var/log/messages`	General system log file.
`/var/mail/<user>`	Default location for the user mailbox.
`/var/run/<inetd>.pid`	File with process ID of the running 'inetd' daemon.
`/var/spool/mqueue/...`	Sendmail queue.
`/var/tmp`	Temporary files; applications should write large files here and not in /tmp, as is usually the case on a larger file system.
`~/.ssh/authorized_keys`	Public keys corresponding to the private keys that can log on to this account using SSH RSA/DAS authentication.

14.1.6 Text Editors

Table 21: Important commands, vi editor.

Command	Description
`:q! [Enter]`	Exit without saving.
`:wq [Enter]`	Write and exit.
`:wq! [Enter]`	Write and exit, forces overwriting of the write-protected file.
`:w filename [Enter]`	Write to another file.
`^L (Ctrl-L)`	Redraw the screen.
`^`	Move to the beginning of the line.
`$`	Move to the end of the line.
`h j k l`	Move the cursor left / down / up / right / up / down (alternative to the arrow keys).
`:num [Enter]`	Go to line number.
`G`	Go to the last line.
`/pattern [Enter]`	Search forward for pattern.
`?pattern [Enter]`	Search backwards for pattern.
`n`	Repeat last search.
`i text ESC`	Insert text before the cursor position.
`A text ESC`	Append text after end of line.
`o text ESC`	Open a new line after the current one and insert text.
`x`	Delete the characters under the cursor.
`r char`	Replaces the character under the cursor with another single character.
`dd`	Delete entire row.
`yy`	Copy ("drag") the current line.
`num yy`	Copying number lines, starting with the current line.
`p`	Paste copy buffer after the current line.

Table 22: Important commands, Easy Editor (ee).

Command	Description
`ESC`	Pop-up menu.
`^C`	Prompt.
`^C quit [Enter]`	Exit without saving.
`^C exit [Enter]`	Write and exit.
`^C write [Enter]`	Write to another file.
`^A`	Move to the beginning of the line.
`^E`	Move to the end of the line.
`^C num [Enter]`	Go to line number.
`^Y string [Enter]`	Search forward for string.
`^X`	Repeat last search.
`^K`	Delete entire row.

14.1.7 Documentation

Table 23: Important commands, documentation.

Command	Description

`man <command>`
`man 5 <command>`
`man -a <command>`
	Displays the manual page for the command <command>. If a page with the same name exists in more than one section, you can specify the section number or -a to display pages from all sections.

`man -k <string>`	Search for string <string> in the manual.
`man hier`	Description of the directory structure
`cd /usr/share/doc; ls` `cd /usr/share/examples; ls`	Browse system documentation and samples. Note in particular /usr/share/doc/de/books/handbook/index.html
`cd /usr/local/share/doc; ls` `cd /usr/local/share/examples`	Browse the package documentation and examples.

14.2 Bibliography

Lucas, Michael W., Absolute FreeBSD: The Complete Guide to FreeBSD, San Francisco 2019.

Lucas, Michael W., Jude, Allan, FreeBSD Mastery: ZFS, Tilted Windmill Press 2015.

The FreeBSD Documentation Project, FreeBSD Handbook, at https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ (accessed 12 July 2019).

Schneider, Wolfram, FreeBSD Manual Pages, at https://www.freebsd.org/cgi/man.cgi (accessed 12 July 2019).

14.3 FreeBSD Copyright

14.4 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com
web:	https://www.beckhoff.com

List of tables

List of figures

More Information:
www.beckhoff.com/twincat-bsd

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com