

Basic Transputer
System Software
for SUN

Version 3.1

June 1995

PARSYTEC reserves the right to make changes in specification at any time and without notice. The information furnished by PARSYTEC in this publication is believed to be accurate, however, no responsibility is assumed for its use, nor for any infringement of patents or rights of third parties resulting from its use. No license is granted under any patents, trademarks or other rights of PARSYTEC.

This manual is Copyright © 1989–1995 by PARSYTEC Eastern Europe GmbH.

Authors: Helmut Blanke-Bohne, Matthias Clauß, Michael Fritzsche, Matthias Rabald, Christoph Steinbrueck, Norbert vom Schemm, Christoph Wiericks

Editor: Helmut Blanke-Bohne

Printed June 13, 1995 in Chemnitz, Germany.

All rights reserved. This document may not, in whole or in part, be copied, photocopied, reproduced or reduced to any electronic medium or machine readable form without any prior consent, in writing, from PARSYTEC Eastern Europe GmbH, Bernsdorfer Str. 210-212, 09126 Chemnitz, Germany.

PARIX™ is a trademark of PARSYTEC GmbH.

UNIX™ is a trademark of AT&T Bell Laboratories in the USA and other countries.

POWERPC™ is a trademark of IBM.

SUN™, SUN-4™, SPARC™ are trademarks of SUN Microsystems, Inc.

X Window System™ is a trademark of the Massachusetts Institute of Technology.

inmos™ and OCCAM™ are trademarks of the INMOS Group of Companies.

The use of registered names, trademarks, etc in this publication does not imply, even in absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Contents

1	Introduction	1
1.1	Changes against previous releases	1
1.2	SunOS / Solaris	2
1.3	Purpose of this manual	2
1.4	Intended audience	2
1.5	Related manuals	3
2	Common Installation	5
2.1	Installation Overview	5
2.2	Requirements	6
2.3	Starting the Software Installation	9
2.4	BTSS specific installation questions	9
2.5	The Configuration File <code>trans_conf</code>	14
2.6	Check the installation	15

3	SCSI-Bus Link Adapter BBK-SCSI	17
3.1	Introduction	17
3.2	Installation	18
3.2.1	BBK-SCSI Specific Requirements	18
3.2.2	Hardware	19
3.2.3	SCSI-Bus Specific Software Configurations	22
3.2.4	Checking the Installation	29
4	S-Bus Link Adapter BBK-S4	33
4.1	Hardware Installation	33
4.1.1	Link Adapter Overview	33
4.1.2	Hardware Requirements	33
4.1.3	Checking the jumper settings	34
4.1.4	Performing the hardware installation	34
4.2	S-Bus Specific Parts for Software Installation	35
4.2.1	Preparing S-Bus device under SUNOS	36
4.2.2	Preparing S-Bus device under SOLARIS	36
5	Vme-Bus Link Adapter	39
5.1	Hardware Installation for VME-Bus SUN's	39
5.1.1	Link Adapter Overview	39
5.1.2	Hardware Requirements	40
5.1.3	Preparing the installation	40
5.1.4	Performing the hardware installation	44

5.2	VME-Bus Specific Parts for Software Installation	46
5.2.1	VME-Bus Specific Requirements	47
5.2.2	VME-Bus Customization of BTSS	47
5.3	Configuration Related Details	48
5.3.1	Description of the VMTM	48
5.3.2	Motivation for the Configuration	48
5.3.3	Description of the Configuring Process	49
5.3.4	Definition of the Topology	51
6	Parsytec Standard Installation	53
6.1	Fast Installation Guide	54
6.1.1	Basic Installation	54
6.1.2	Reinstallation	54
6.2	Detailed Installation Guide	55
6.2.1	Requirements for Installation	56
6.2.2	Read the Tape	57
6.2.3	Checking the Delivery	58
6.2.4	Product Specific Preparations	59
6.2.5	Starting the PARSYTEC Standard Installation	61
6.2.6	Installation Program	62
6.2.7	User Interface	62
6.2.8	Reinstallation	67
6.2.9	Installation Stuff inside the Product Directory	68
6.3	Installations on Additional Workstations	68
6.4	Installing a Patch Set	69
6.5	Difficulties	70

- 7 BTSS Tools** **71**

 - 7.1 links 71
 - 7.2 ulti 72
 - 7.3 Tools for BBK-SCSI 73
 - 7.3.1 getversion 73
 - 7.3.2 burn_scsi 74
 - 7.4 Tools for BBK-S4 74
 - 7.4.1 s4diag 75
 - 7.5 Tools for VME-Bus 75
 - 7.5.1 vmtmconf 75

- 8 Public Link Interface** **77**

 - 8.1 Internal Information 78
 - 8.2 Link Interface Function Description 79

- 9 Troubleshooting** **89**

 - 9.1 No Light on Front LEDs (BBK-SCSI) 89
 - 9.2 Error Message: Failed to get existing sem_key 90
 - 9.3 Errors while Reading the Config File 90
 - 9.4 Errors within the Description of Your Hardware 91
 - 9.5 Error Messages about Incorrect Parameters 92
 - 9.6 A Link is Showed as 'in use' but Nobody Uses it 92
 - 9.7 Error Message: Access to link hardware failed 93

- 10 Tips and tricks** **95**

- Index** **97**

Chapter 1

Introduction

The Basic Transputer System Software (BTSS) is the driver software for link adapter, the interface between SUN and parallel machines. It has to be installed for every SUN containing a link adapter.

BTSS version 3.1 supports the link adapter types BBK-SCSI (sometimes also called SCSI-T), BBK-S4 and the VME-Bus link adapter BBK-V2, BBK-V4 and VMTM.

This manual includes the hardware (installation) description for the above types of link adapter.

1.1 Changes against previous releases

The BTSS version 1.3 supported the VME-Bus and the S-Bus link adapter for SunOS 4.x. SOLARIS support came with version 2.11. This version (3.1) also supports the new PARSYTEC link adapter BBK-SCSI.

The host adapter BBK-V1 and MTM-Sun are no longer supported.

1.2 SunOS / Solaris

The meaning of SUNOS and SOLARIS may be a little bit confusing: SunOS is usually used for the operating system version SunOS 4.x, which is sometimes called Solaris 1.x too. Solaris stands for Solaris 2.x, also known as SunOS 5.x.

Throughout this manual, **SunOS** is used if **SunOS 4.x** is meant and **Solaris** if **Solaris 2.x** is meant.

1.3 Purpose of this manual

The purpose of this manual is to describe the Basic Transputer System Software, how to install it, how to change the default topology established on boards with electronic link-switches, and the interface to transputer hardware. In general installing a development system from PARSYTEC requires the following steps:

- Install the hardware
- Install the Basic Transputer System Software (BTSS)
- Install the development system

1.4 Intended audience

The hardware installation part is strongly related to the administrator of the transputer subsystem. It gives all the information necessary for a successful installation.

Before installing the software the specific bus needs to be configured. This step is different for the available bus systems. It is described in the section

'bus Specific Parts for Software Installation' (cf SCSI-Bus: 3.2.3, S-Bus: 4.2, VME-Bus: 5.2).

The software installation is nearly the same for all bus systems and is done via the PARSYTEC Standard Installation (cf chapter 6). During the installation superuser permission is required.

For boards containing electronic link-switches (e.g. VMTM) a brief description is given for changing their default topology according to local needs.

The description of the link interface is intended to be used by interested programmers, who want to access a link directly. A detailed description is given to enable writing standalone applications.

1.5 Related manuals

- For details of the hardware see the technical documentation of the BBK-V2, BBK-V4, VMTM, BBK-S4 and BBK-SCSI boards.
- See SUN's "Card Cage Slot Assignment And Backplane Configuration Procedure" for more information about slot usage and jumper locations of VME-Bus based systems.
- The installation of software development product like TOOLSET, PowerTools or PARIX is described in the according delivery manuals.

Chapter 2

Common Installation

The installation of the link adapter hardware is different for the various types of link adapter. The hardware installation steps are described in the bus specific chapters (SCSI-Bus: chapter 3, S-Bus: chapter 4, VME-Bus: chapter 5).

The software installation is similar for the different types. Therefore this chapter describes the BTSS installation for all link adapter types and refers to the link adapter specific sections if specific things have to be done.

It also refers to the PARSYTEC Standard Installation (chapter 6) which is a common PARSYTEC tool for software installation.

2.1 Installation Overview

The installation steps are:

- Install the hardware (cf hardware installation sections for the specific link adapter type, i.e. SCSI-Bus: section 3.2, S-Bus: section 4.1, VME-Bus: section 5.1).

- Check the common installation requirements for the PARSYTEC Standard Installation (cf section 6.2.1), the BTSS specific requirements (section 2.2) and the specific requirements for the specific link adapter type (SCSI-Bus: section 3.2.1, S-Bus: section 4.2, VME-Bus: section 5.2.1) .
- Read the tape via PARSYTEC Standard Installation (cf section 6.2.2)
- Control the delivery via PARSYTEC Standard Installation (cf section 6.2.3)
- Start the PARSYTEC Standard Installation to install BTSS (cf section 6.2.5)
- Answer product specific questions for BTSS using section 2.4 and 6.2.7.
- Do the first installation step for BTSS installation (cf sections 6.2.5, 6.2.6 and 6.2.7)
- Create or adapt the BTSS configuration file `trans.conf` located on the BTSS root directory (second installation step).
- Only for SCSI-Bus: Do the SCSI-Bus specific software configurations listed in section 3.2.3 and switch on the BBK-SCSI.
- Do the third installation step via reinstallation in the BTSS root directory (cf section 6.1.2 or 6.2.8) and answer the BTSS specific question using the second part of section 2.4.
- Only for SCSI-Bus: Switch off and on again the BBK-SCSI in order to activate the firmware just burned in.

2.2 Requirements

Please check first the standard requirements for the PARSYTEC Standard Installation described in the section 6.2.1. The PARSYTEC Standard Installation will be used to install BTSS.

Following BTSS specific requirements are needed, login as superuser to check them:

- The host machine must be a SPARCSTATION or a SPARCSEVER running SUNOS 4.1.3 (or higher) or SOLARIS 2.3 (or higher). It may be checked by:

```
uname -a
```

- The interprocess communication facilities `ipcs` must be installed. BTSS uses **semaphores** to guarantee exclusive access to a link of a link adapter. The standard generic kernel includes this feature, but sometimes semaphores are removed from locally adapted kernels, frequently. Check the presence of semaphore support by the command:

```
ipcs -sb
```

If `ipcs` reports missing semaphores you have to reconfigure your kernel. An administrator who has built this adapted kernel, should know how to include semaphores.

- The required disk space for the installed Basic Transputer System Software is about 1.5 MByte. Additional space (1.5 MByte) is needed at a temporary location.
- The Basic Transputer System Software (BTSS) for SUN-4 is delivered in tar format on a single 3.5" floppy disk or together with other software products on a 1/4" streamer tape readable on a QIC-150 tape device. The PARSYTEC Standard Installation is delivered on the same floppy disc or tape.
- The default location for the BTSS tree is

```
/usr/etc/transp
```

Owner and group should be `bin` and `vmebus`. We strongly recommend not to use another group than `vmebus`.

- Check the existence of a group *vmebus* by using:

```
more /etc/group
```

or

```
ypcat group | grep vmebus
```

To check if this group was entered into the NIS database look for the line

```
vmebus:*:25:
```

If this line is not included in your */etc/group* file or your NIS database you must add this line to the file */etc/group* on all machines which shall become connected to a link adapter.

The **user group** *vmebus* must have read/write permission to the link adapter. It is strongly recommended to create this group named *vmebus* with the numeric gid 25.

If you are using the yellow-page services for group membership, make an appropriate entry. Include the following line into the file */etc/group*:

```
vmebus:*:25:parix,toolset:
```

Comment: In order to clarify the ownership of files the transputer software development tools Parix and Toolset use dummy users named *parix* and *toolset*. Other development tools may require additions to the group entry.

Don't forget to check the requirements for the specific link adapter type, cf SCSI-Bus: section 3.2.1, S-Bus: section 4.2, VME-Bus: section 5.2.1.

2.3 Starting the Software Installation

Read the tape or floppy disc and start the installation process by following the PARSYTEC Standard Installation documentation (cf section 6.1 or 6.2) and answer the product specific questions by using the following section (2.4). For an installation overview see section 2.1.

2.4 BTSS specific installation questions

During the installation of BTSS via the PARSYTEC Standard Installation product specific questions are asked. This section gives some hints to answer them. Note that the installation has to be done in three steps.

The first step installs the BTSS and generates an example file `trans_conf` which must be adapted to your physical link adapter configuration on this SUN (second step).

After you have updated the file `trans_conf` (cf section 2.5), you can do the third installation step by reinstallation (cf section 6.1.2 or 6.2.8). The items of the first installation step need not be done again, the `trans_conf` example generation must not be done again because it then overwrites your adapted file.

BTSS questions in the first installation step are:

- The location where to install the BTSS software:

```
PATH to install BTSS : We recommend to install it in the directory
'/usr/etc/transp'.
```

```
Enter the (full) path name for 'Basic Transputer System Software for Sun'!
Where do you want to install it ?:
(/usr/etc/transp)
```

The BTSS software usually is installed in the directory `/usr/etc/transp`. If you want to install it in another location create a symbolic link `/usr/etc/transp` to the location where you install the software.

Enter 'RET' to accept the standard path.

- Owner for the BTSS:

We recommend to use the owner bin for BTSS.

- Group for the BTSS:

GROUP : BTSS needs the group 'vmebus'. (It must be a group with read/write permission to S-Bus and/or SCSI-Bus.)

Enter the group name for 'BTSS' (vmebus) :

The group vmebus (created before) has to be used. In former times it described the group with access to the VME-Bus. Today it includes also the access to S-Bus and SCSI-Bus. The name of the group is fixed.

Enter 'RET' to accept the standard group.

- Access for others for BTSS:

Choose 'access for others', but BE CAREFUL: Removing the access for others means that only users belonging to the 'vmebus' group get access to the parallel machine via link. (We strongly recommend to answer 'yes'.)

Give access rights to others for BTSS ? (y/n) :

Enter 'yes' for this question.

- Bus type of link adapter:

VME-Bus : The next two questions are only relevant for VME-Bus link adapter. Answer 'no' if you only have BBK-S4 and/or BBK-SCSI.

If you have a VME-Bus link adapter you have to know whether the VME-Bus is 24 or 32 bit. Answer only one of the following two questions by 'yes'!

Install for 24 bit VME-Bus? (y/n) :

n

Install for 32 bit VME-Bus? (y/n) :

y

Install for S-Bus ? (y/n) :

y

Install for SCSI ? (y/n) :

y

Answer the respective question for the link adapters in your SUN by 'yes'. If you have a mixed installation you may answer up to three questions by 'yes'. The only restriction is that you may only answer one VME-Bus question by 'yes' otherwise one answer will be overwritten.

VME-Bus with 24 bit may occur for SUN 4-110 or SUN 4-140. All other VME-Bus machines have 32 bit.

- BTSS tools may be copied to the common binary directory `/usr/bin` to give access for the tools `links` and `ulti` for anyone:

COPY BTSS TOOLS : The tools 'ulti' and 'links' should be available to anyone. They may be copied to `/usr/bin` by this installation. (You may also COPY them MANUALLY to another directory included in the user's path. After installation they are located in the root directory of BTSS. Note that these tools must belong to owner 'bin' and group 'vmebus' and the 'Set Group Id' bit has to be set for the group.)
Copy BTSS tools (ulti, links) to `/usr/bin` ? (y/n) :

The `links` command lists all links to transputer hardware together with their availability. The `ulti` command checks and optionally configures all transputers. Its intended use is a quick check of the transputer after power up and establishing the static topology for boards featuring link-switches. The `ulti -h` command gives a usage message.

During the installation you get the possibility to copy these tools to the directory `/usr/bin`. If you don't want them located there answer the question by 'no' and follow the manual copy instructions in the above question.

- The BTSS S-Bus tool `s4diag` may also be copied to the common binary directory `/usr/bin`:

Copy BTSS S-Bus tools (s4diag) to `/usr/bin` ? (May be done manually afterwards, cf. COPY BTSS TOOLS.) (y/n) :

`s4diag` is an administrator tools to control the BBK-S4. If you don't use S-Bus answer the question by 'no'.

During the installation you get the possibility to copy this tool to the directory /usr/bin. If you don't want it located there answer the question by 'no' and follow the the manual copy instuctions in the question for uli and links.

- Generate an example trans_conf file:

```
EXAMPLE FILE 'trans_conf' : A file describing the hardware configuration
named 'trans_conf' is needed in the BTSS root directory. It has to be
adapted to the actual hardware configuration after the first installation
step. You may install an example file within this installation.
(You may type 'yes' in the first installation step to get a file with
syntax definition and example entries. Type 'no' if you want to KEEP YOUR
OLD 'trans_conf' file, especially if you already updated it before. )
Install an example 'trans_conf' file ? (y/n) :
```

The trans_conf is the file to configure link adapters in the SUN. Each SUN has its own trans_conf. For each link adapter in the SUN one line entry must be contained in this file.

Within the first installation anwer this question by 'yes'. You get an example file which you can edit after the first installation step. See also section 2.5.

- Skip installation questions for the third step:

... (step 3) ...

Answer the questions indicated by (step 3) with 'no' in the first installation step. These items must not be done in the first step because they need the trans_conf file up to date.

Before you do the third installation step adapt the file trans_conf in the BTSS root directory. The third installation step is started by a reinstalation step via the PARSYTEC Standard Installation (cf section 6.1.2 or 6.2.8).

BTSS questions in the third installation step ¹ (reinstallation step) are:

¹If both questions for step 3 (firmware and semaphore) will be answered by 'no' the third installation step may be skipped.

- Skip installation questions for the first step:

After you entered owner, group and 'other access' answer the questions not indicated by (step 3) with 'no'.

- Burn in the firmware for BBK-SCSI:

BURN IN BBK-SCSI FIRMWARE (step 3) : The BBK-SCSI devices may get a burn in with the actual firmware during installation. BE SURE that the configuration file 'trans_conf' in the BTSS root directory has been adapted to the actual hardware.

(Answer the question below by 'no' in the first installation step. Adapt the file 'trans_conf' at the BTSS root directory. Call 'make install' again in the BTSS root directory and type 'yes' for the question below.)
Are you sure 'trans_conf' is up to date and do you want to burn in the actual firmware for BBK-SCSI ? (y/n) :

Answer 'no' if you only work with S-Bus and/or VME-Bus.

Answer 'yes' only if you have a BBK-SCSI installed in your SUN and if an entry line for BBK-SCSI (indicated by SCSI-T) exists in the configuration file trans_conf.

If a burn in once has been done for a BBK-SCSI you need not repeat it until a firmware update is available.

Do not forget to reset the BBK-SCSI by switching off and on again, otherwise the new firmware will not become active!

- Semaphore initialization:

INITIALIZE SEMAPHORE (step 3) : After the configuration file 'trans_conf' has been adapted to the actual hardware the link semaphore has to be initialized.

(Answer the question below by 'no' in the first installation step. Adapt the file 'trans_conf' at the BTSS root directory. Call 'make install' again in the BTSS root directory and type 'yes' for the question below to initialize the semaphore.)

Is the file 'trans_conf' in the BTSS root directory up to date ? (y/n) :

The first access to the semaphore initializes it. If an arbitrary user does the first access the semaphore usually does not belong to the group vmebus. The effect is that other users don't have access to

the semaphore and in consequence they have no access to the link hardware.

To omit this effect do the initialisation of the semaphore in the third installation step by answering the above question by 'yes'.

You may control the existence of the semaphore before the third step by `ipcs -sb`. If the semaphore exists with the key `0x42424242` and belongs to the group `vmebus` it is ok and need not be initialized again. If a semaphore with that key exists and does not belong to the group `vmebus` delete it by `ipcrm -s semid` and answer the above question by 'yes' in the third installation step.

2.5 The Configuration File `trans_conf`

The configuration file `trans_conf` in the BTSS root directory (`/usr/etc/transp`) has to contain a description line for any link adapters in this SUN.

The installation process generates a `trans_conf` example file in it's first step. (Answer the repetitive question by 'no' if you want to keep your old `trans_conf`).

The generated file contains a syntax description and some example lines for the different link adapter types.

Some lines are generated in dependency of your answers during the installation process. Please control (and adapt) the lines to get the right configuration.

If you already have installed PARSYTEC software, we advise you to keep the already existing `trans_conf` file in your private directory in order to copy it back to it's location if something went wrong.

If you have more than a single link adapter one line is needed for each. Lines starting with '#' are regarded as comment lines. Outcomment the lines according to your local configuration.

For example, let us assume you have a BBK-V4 and a VMTM board without additional NCU (network configuration unit). During the hardware

installation you have decided to use the following addresses:

```
BBK-V4 $90000000
```

```
VMTM $80000000
```

There should be two lines inside the `trans.conf` file reading:

```
0 BBKV4 90000000
```

```
1 VMTM 80000000
```

An example and further hints for BBK-SCSI you may find in the SCSI-Bus specific section 3.2.3.3.

2.6 Check the installation

Check the complete installation by calling `ulti -v`.² Be sure that a transputer is attached to every Lemos connector at the link adapter's links, otherwise `ulti` will complain with:

```
link not connected to a transputer
```

For the first time you must update your shell hashtables by

```
rehash
```

Now try the command

```
ulti -v
```

²Note, that `ulti` will be in your path only if you answered the question for copying BTSS tools to `/usr/bin` by 'yes'. Otherwise you have to make `ulti` available in another way. For the check you might call `/usr/etc/transp/ulti`.

You will see a lot of diagnostic messages informing you in detail about the actions taken by `ulti`, here an example for two link adapter installed, a BBK-S4 and a BBK-SCSI:

Hi there! Welcome to the ultimate transputer board test-utility.

The config file tells me the following configuration is available on your workstation.

Id	Status	Link-Name	Bus-Address
0	available	bbk-s4 b0 10	0xfe000100
1	available	bbk-s4 b0 11	0xfe000480
2	available	bbk-s4 b0 12	0xfe000800
3	available	bbk-s4 b0 13	0xfe000b80
4	available	scsi-t b1 10	/dev/rst0
5	available	scsi-t b1 11	/dev/rst1
6	available	scsi-t b1 12	/dev/rst2
7	available	scsi-t b1 13	/dev/rst3

number of used links: 0

```
Testing link "bbk-s4 b0 10" ... FAILED!
  Link not connected to a transputer (board not configured)
Testing link "bbk-s4 b0 11" ... FAILED!
  Link not connected to a transputer (board not configured)
Testing link "bbk-s4 b0 12" ... OK
Testing link "bbk-s4 b0 13" ... FAILED!
  Link not connected to a transputer (board not configured)
Testing link "scsi-t b1 10" ... FAILED!
  Link not connected to a transputer (board not configured)
Testing link "scsi-t b1 11" ... FAILED!
  Link not connected to a transputer (board not configured)
Testing link "scsi-t b1 12" ... OK
Testing link "scsi-t b1 13" ... FAILED!
  Link not connected to a transputer (board not configured)
```

Have tested 2 of 8 transputers successfully

Note, that only link 2 (board b0 link 12) and link 6 (board b1 link 12) were connected to a transputer in this example. The corresponding line entries in the file `trans.conf` are:

0	BBKS4	3	BBK-S4 board in SBus slot 3
1	SCSI-T	/dev/rst0	SCSI-T device for SunOS 4 (uses rst0 to rst3)

Chapter 3

SCSI-Bus Link Adapter BBK-SCSI

3.1 Introduction

This paper describes the hard- and software of the BBK-SCSI. It explains how to install and to use the BBK-SCSI and the appropriated Basic Transputer System Software (BTSS).

The purpose of the BBK-SCSI is to connect any PARSYTEC parallel machine with transputer link interface like a GC/PowerPlus or POWERXPLO- RER to any host machine capable of using SCSI tape devices. The current version supports SUN host machines. Therefore, this paper deals especially with the connection to these machines.

The BBK-SCSI consists of a hardware part containing in a box mainly a transputer, a SCSI-chip and cache memory, and a BTSS packet providing you with some object files and tools for testing and using the BBK-SCSI.

While using the widely provided SCSI standard you don't need an especially designed card for each architecture anymore. Using the standard

SCSI bus guarantes you low costs and compatibility to all major companies computers. The intelligent BBK-SCSI firmware uses a cache into both directions and supports the SCSI-2 standard to improve the transfer performance.

The upgradable firmware allows to implement different and new communication protocols on your transputer link interface. Currently it supports the PARIX standard communication protocols for T8-PARIX and POWERPC-PARIX as well as the Streammode standard, e.g. used with the Iserver of TOOLSET.

3.2 Installation

3.2.1 BBK-SCSI Specific Requirements

Before starting the BBK-SCSI installation please check the following facts on the machine:

- The tape driver on this machine must exist.
- For each link you are intending to use an unused minor-device of the machine's tape driver must exist. SUNOS supports only 8 minor devices for the tape driver. If you want to use two BBK-SCSIs on your machine and you want to use all 8 links you can't use a tape drive on this machine. If you use only one BBK-SCSI you can still use up to four tape drives. Running SOLARIS you don't have this limit. The configuration of your machine for an appropriated number of minor devices of the tape driver is described in section 3.2.3.

We recommend not to use more than two BBK-SCSIs on one SUN (for perfomance reasons).

- You need a free connector of an SCSI-bus with an unused SCSI-id for each BBK-SCSI on the choosen host machine. The connector

should be a 50-pin high density standard connector (on the host adapter).

If you don't have SCSI support yet you can order a dedicated SCSI host adapter.

It is possible to use the BBK-SCSI on every existing SCSI host adapter with a free SCSI-id. However, for achieving the best possible performance you should use a dedicated SCSI host adapter for the BBK-SCSI. This host adapter must be a single ended SUN SCSI Adapter Fast-SCSI-2 (10 MByte/s).

- For each first BBK-SCSI on one host you need a host adapter cable for SUN (high density 50 to Centronics 50, length: 1.5 meter) For each second BBK-SCSI you need a SCSI device connection cable (Centronics 50 to Centronics 50, length: 0.5 meter). If your host adapter uses a different connector (not high density 50 pin) you need an appropriated cable but with a Centronics 50 connector on the other end.
- For each host you need an active terminator (Centronics 50).
- For each link you want to use you need a LNK-10 link adapter cable (normally 4 per BBK-SCSI).

The installation requires superuser permission.

3.2.2 Hardware

3.2.2.1 Technical Data

Dimensions: 244mm * 70mm * 265mm (w*h*d)

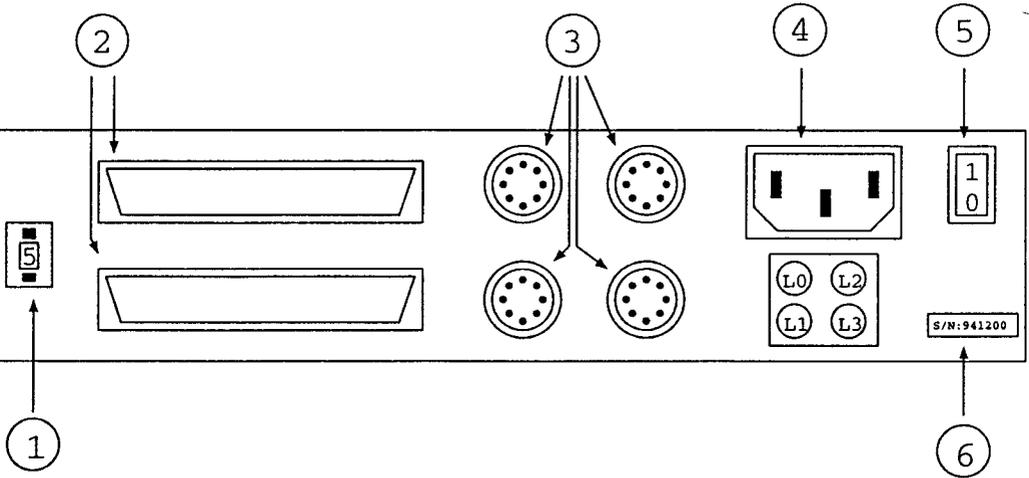
Power requirements

Voltage : 220 V +/- 20%

Consumtion : 60 W

The maximum transfer rate for one link is: 1200 kByte/s.

Switches and connectors on the backplane:



- 1 : SCSI-id switch
- 2 : 50-pin centronics SCSI-bus connectors (in and out)
- 3 : 8-pin Lemosa-connectors for the four links
- 4 : power line connector
- 5 : power switch
- 6 : serial number

Jumpers (inside):

number	position	description
1	1-2	boot from Flash-PROM (default)
	2-3	boot from link (service only)
2	1-2	default (don't change!)
3	1-2	default (don't change!)
4	1-2	default (don't change!)
5	2-3	default (don't change!)
6	1-2	default (don't change!)
7	open	default (don't change!)
8	open	disable link-in-reset (default)
	1-2	enable link-in-reset (service only)
9	open	disable link-in-reset (default)
	1-2	enable link-in-reset (service only)
10	open	disable link-in-reset (default)
	1-2	enable link-in-reset (service only)
11	open	disable link-in-reset (default)
	1-2	enable link-in-reset (service only)

3.2.2.2 Installing the Hardware

Attention! All changes and connections concerning your host machine and the BBK-SCSI should be done after powering off the machines!

Installing the hardware of the BBK-SCSI is easy. You connect the BBK-SCSI with a standard 50-pin centronics or DB-50 cable to your machine. Pay attention for correctly installed SCSI terminators. Each end of the SCSI-bus needs to be terminated on the appropriated device. All other devices must not have a terminator.

Choose a free SCSI-id for your BBK-SCSI and adjust the number on the SCSI-id switch on the backplane of the BBK-SCSI accordingly. For choosing possible SCSI-id please refer also to section 3.2.3. For later changes of the id please note: a change takes only effect if you turn the BBK-SCSI off and on.

Connect the link connectors 0-3 with the appropriated link connectors of your PARSYTEC system (e. g. GC/PowerPlus). Finally connect the power line.

If you turn on the BBK-SCSI the red and the green LED on the front pannel should light up for a short time. After that, the red LED should turn off, the green one should light up about three times a second. If your BBK-SCSI doesn't show this behavior please conctact the service. Otherwise your BBK-SCSI is now ready for using it.

3.2.3 SCSI-Bus Specific Software Configurations

To do the software installation process refer to chapter 2. That section will refer to this section to do SCSI-Bus specific installation steps.

The SCSI-Bus specific configuration has to be done after the first and second BTSS installation steps (c f Installation Overview in section 2.1).

The third BTSS installation step (burn in firmware and initialize the semaphore) has to be done afterwards.

The SCSI-Bus specific software configuration consists of several steps:

1. Design the spreading of your SCSI-ids¹ between your SCSI-devices.
2. Configure your system for using this spreading of the SCSI-ids and devices and connect your BBK-SCSI(s).
3. With the delivered tools `links` and `getversion`, check the connection of the BBK-SCSI(s) and the configuration of your system.

These steps are described in the following sections.

¹SCSI-id: each device attached to a SCSI-bus must have an unique id for this SCSI-bus. You can choose this id by setting some jumpers or setting a switch. Refer to the description of your SCSI-devices. The BBK-SCSI has a switch on the backplane (refer to section 3.2.2.1).

3.2.3.1 Designing of the SCSI-ids

As described in section 3.2.1 you need to configure your SCSI-devices in a manner that you have as many free SCSI-ids on the desired SCSI-bus as many BBK-SCSIs you want to use. The following suggestion for a partition of your SCSI-id's is based on the common manner to use SCSI-id 4 and 5 for tape devices. The id 7 is reserved for the SCSI hostadapter. Note: if you have more than one SCSI-busses, you can use the same ids but on different SCSI-busses.

example configuration with no tape drives and one BBK-SCSI:

SCSI-id	used for
0-3	SCSI hard discs, other devices or free
4	BBK-SCSI 0
5,6	SCSI hard discs, cdroms, other devices or free
7	SCSI host adapter

example configuration with one tape drive and one BBK-SCSI:

SCSI-id	used for
0-3	SCSI hard discs, other devices or free
4	SCSI tape drive
5	BBK-SCSI 0
6	SCSI hard discs, cdroms, other devices or free
7	SCSI host adapter

example configuration with no tape drives and two BBK-SCSI:

SCSI-id	used for
0-3	SCSI hard discs, other devices or free
4	BBK-SCSI 0
5	BBK-SCSI 1
6	SCSI hard discs, cdroms, other devices or free
7	SCSI host adapter

After you made sure about the partitioning of the SCSI-id's and checking for conflicts with already used SCSI devices you can go ahead with configuring your system.

3.2.3.2 Configuring your System and connecting the BBK-SCSIs

As described in section 3.2.1 you need to configure your *operating system* in order to have as many free minor numbers of the systems tape-driver as many links via the BBK-SCSIs you want to use (usually 4 times number of BBK-SCSIs per machine). In order to fullfill this requirement, you need to change the configuration of your host in one of the following ways, depending if your host runs SUNOS or SOLARIS. In both cases, this requires a reboot of your machine, so ensure yourself if this is possible!

3.2.3.2.1 Configuration for hosts running SunOS

- Become superuser.
- Change the working directory to the machine's configuration description directory. For SUN machines this is usually a directory `/usr/kvm/sys/sun4?/conf`, there `?` is a letter arcodingly to your current used operating system (usually `c` or `m`).
- If you entered the right directory you should find a file named `GENERIC`, which contains besides other data the machine's configuration for the SCSI-id's. If you are already use a different configuration you should start from the most recently used copy of the configuration file. Make a copy of this file with the name `PARSYTEC`. With any character editor you need now to edit the section of the SCSI device units. This section is headlined by:

The following section describes SCSI device unit assignments.

Choose the section with the appropriated SCSI host adapter connected to your BBK-SCSI. The different busses are called *scsibus0*, *scsibus1* and so on. For the second SCSI-bus it would be headlined as follows:

```
scsibus1 at esp                # declare second scsi bus
```

You need to add lines like the following:

```
tape st0 at scsibus1 target 4 lun 0 # Parsytec SCSI-T8-Bridge
tape st1 at scsibus1 target 4 lun 1 # Parsytec SCSI-T8-Bridge
tape st2 at scsibus1 target 4 lun 2 # Parsytec SCSI-T8-Bridge
tape st3 at scsibus1 target 4 lun 3 # Parsytec SCSI-T8-Bridge
```

where is the number after target the SCSI-id and the number after lun an indicator for link 0 to 3. Of course, you also could use scsibus0 or scsibus2 and so on.

Take care that no other entries define the same devices (tape st0 - tape st3 in this case). If they do uncomment these lines. Make sure that all other entries in this file refer to the current configuration of your machine.

- Run the program `/etc/config` with the parameter `PARSYTEC` as following:

```
/etc/config PARSYTEC
```

- Change the current directory to the new generated directory `../PARSYTEC` typing the following line:

```
cd ../PARSYTEC
```

and run make by typing

```
make
```

- Make a security copy of your old *operating system* by typing

```
cp /vmunix /vmunix.old
```

- Copy the new *operating system* into the root directory with

```
cp vmunix /vmunix
```

- Bring the machine down. For example, use the following command:

```
shutdown -h +5
```

- This is the perfect moment to turn your machine of and to connect your BBK-SCSIs to your host. Use the high density/Centronics cable to connect the first BBK-SCSI to your host adapter and the Centronics/Centronics cable to connect a further BBK-SCSI. The last BBK-SCSI must be terminated with the required active Centronics SCSI-terminator (except there are following some other devices).
- Check the connection of your BBK-SCSIs and turn them on. Reboot the machine to put the changes into effect.

```
boot
```

If your BBK-SCSI was turned on and all changes made correctly you should see outputs as follows (besides others):

```
st0 at esp1 target 4 lun 0
st0: <Vendor 'Parsytec' Product 'SCSI-T8-BRIDGE' >
st1 at esp1 target 4 lun 1
st1: <Vendor 'Parsytec' Product 'SCSI-T8-BRIDGE' >
st2 at esp1 target 4 lun 2
st2: <Vendor 'Parsytec' Product 'SCSI-T8-BRIDGE' >
st3 at esp1 target 4 lun 3
st3: <Vendor 'Parsytec' Product 'SCSI-T8-BRIDGE' >
```

The *operating system* of your machine is now configured for using the BBK-SCSI.

3.2.3.2.2 Configuration for hosts running Solaris

- Become superuser.

- Change the working directory to the machine's built-in configuration table directory (*'/kernel/drv/'*). The file *'st.conf'* contains all possible tape devices. Usually, this description file contains only one logical unit (lun) per target. For configuring the system for the BBK-SCSI you need to declare four logical units per target.
- We consider to extend at least target 4 and 5 for using BBK-SCSIs. You can change both targets even if you want to use only one BBK-SCSI at first. A change for other targets does not harm your system. You will find a section like this:

```
name="st" class="scsi"
target=4 lun=0;
```

You need to add the following lines:

```
name="st" class="scsi"
target=4 lun=1;
```

```
name="st" class="scsi"
target=4 lun=2;
```

```
name="st" class="scsi"
target=4 lun=3;
```

For target 5 please do the appropriated changes.

- Change to the directory *'/dev/rmt'* and remove all links pointing to the target on the SCSI-bus you want to use for the BBK-SCSI inside of this directory. For target 4, for example, they point to device nodes like *st@4,0*, *st@4,1* and so on. The internal SCSI-bus is located in a directory like *esp@0,80000*, an additional SCSI-bus is located in a directory like *esp@f,800000*.

The easiest and most secure case would be, if you remove all links inside of this directory. For this case, during the following reboot

you need to connect and power on all tape drives you want to use further. This ensures a correct installation.

- Bring your machine down. For example, use the following command:

```
shutdown -y -i0 -g300
```

- This is the perfect moment to turn your machine of and to connect your BBK-SCSIs to your host. Use the high density/Centronics cable to connect the first BBK-SCSI to your host adapter and the Centronics/Centronics cable to connect a further BBK-SCSI. The last BBK-SCSI must be terminated with the required active Centronics SCSI-terminator (except there are following some other devices).
- Check the connection of your BBK-SCSIs and turn them on. Boot your machine with the 'reconfigure'-flag.

```
boot -r
```

The machine will now determine the new devices and create the necessary device nodes.

The *operating system* of your machine is now configured for using the BBK-SCSI.

3.2.3.3 Configuring and Checking the BBK-SCSIs

To be able to access the BBK-SCSI the configuration file must have been set up (cf second step of BTSS installation, see section 2.1). This file describes the hardware of your SUN used to get the connection to the parallel machine. In this case it is the BBK-SCSI.

This file is situated in the directory where you have installed the BTSS to. Usually it is the directory `/usr/etc/transp`. For using BBK-SCSIs as well as any other hardware like BBK-S4 or BBK-V2 you need to adapt the file `trans.conf` inside of this directory. For each BBK-SCSI it has to contain an entry as follows

for SUNOS:

```
0      SCSI-T /dev/rst0      SCSI-T device for SunOS 4 (uses rst0 to rst3)
1      SCSI-T /dev/rst4      SCSI-T device for SunOS 4 (uses rst4 to rst7)
```

or for SOLARIS:

```
0      SCSI-T /dev/rmt/0      SCSI-T device for Solaris (uses ../0 to ../3)
1      SCSI-T /dev/rmt/4      SCSI-T device for Solaris (uses ../4 to ../7)
```

The first entry describes the unique board number for this host. The second entry describes the type of the hardware, in this case a BBK-SCSI. The third entry needs to correlate with the device node for the first logical unit of the target which represents the BBK-SCSI. The examples showed above are valid for configurations without any other tape devices. If you use tapes additional to the BBK-SCSI the names of the device nodes may differ.

If you have made this entries, you should check if your BBK-SCSIs are working. If you have the right entries in your file `trans_conf`, the tool `links` should show you all links (with the Status available) (call simply `links` without any arguments). With the tool `getversion`² you can proof, if your configuration is working and your BBK-SCSIs are accessible. This tool should give you an information about the firmware version inside of the BBK-SCSI (call `getversion` with the devicename (s) you have entered into the file `trans_conf`).

Please, refer to the sections 7.1 and 7.3.1 to get more information about this tools. **Do not use the tool `ulti` at this time during the installation!** If you get some error messages, refer to section 9.

3.2.4 Checking the Installation

The installation of the BTSS should have created the following files (besides others):

²`getversion` is located in the subdirectory `scsi_tools/SunOSx` of the BTSS root directory (x is 4 for SUNOS 4 and 5 for SOLARIS).

1 available scsi-t b1 l3 /dev/rst7

number of used links: 0

Testing link "0" ... FAILED!

Link not connected to a transputer (board not configured)

Testing link "1" ... FAILED!

Link not connected to a transputer (board not configured)

Testing link "2" ... FAILED!

Link not connected to a transputer (board not configured)

Testing link "3" ... FAILED!

Link not connected to a transputer (board not configured)

Testing link "4" ... FAILED!

Link not connected to a transputer (board not configured)

Testing link "5" ... FAILED!

Link not connected to a transputer (board not configured)

Testing link "6" ... FAILED!

Link not connected to a transputer (board not configured)

Testing link "7" ... FAILED!

Link not connected to a transputer (board not configured)

Have tested 0 of 8 transputers successfully

or, if all links are connected to transputers:

Testing link "0" ... OK

Testing link "1" ... OK

Testing link "2" ... OK

Testing link "3" ... OK

Testing link "4" ... OK

Testing link "5" ... OK

Testing link "6" ... OK

Testing link "7" ... OK

Have tested 8 of 8 transputers successfully

If you get some errors, check at first all steps from section 3.2.3 once more and than read the section 7 to check the correct configuration of your BBK-SCSI.

Chapter 4

S-Bus Link Adapter BBK-S4

This chapter describes the S-Bus specific installation steps. S-Bus link adapter is the BBK-S4.

4.1 Hardware Installation

4.1.1 Link Adapter Overview

For S-Bus based SUN workstations PARSYTEC provides the BBK-S4 board, which requires a single S-Bus slot.

4.1.2 Hardware Requirements

In order to use BBK-S4 boards, your SUN must have a free S-Bus slot.

4.1.3 Checking the jumper settings

It is advised to check the BBK-S4 jumper settings. Each board is carefully checked before delivery. This check includes the factory default jumper settings. But due to the human nature errors may occur.

4.1.4 Performing the hardware installation

The installation of the hardware requires the computer to be powered down. Power down the system.

Hint:

Software installation under Solaris 2.x requires a system shutdown as well (see 4.2.2). Therefore you may continue with preparation of the BTSS software installation described in chapter 2

But be carefully :

Hint:

If you have any doubts about knowing exactly how to power down the system, ask your system manager. The filesystem may be damaged if you do not obey the correct powerdown procedure!

- Power down your system. Be sure that no other users are logged in or they will be disturbed by your power down. This may happen if this system is used as a file server.

Usually the command

```
shutdown -h
```

will work fine. After some time the message

```
unix halted
```

should appear. Otherwise contact your local system manager. Now you can switch off the power-supply of the computer. Remove the power cable from the computer for security reasons.

- Remove the cover from your S-Bus SUN.
- Insert the S-Bus board into the chosen SUN slot.
- Remount the cover of your SUN, reconnect the power cable and switch on the computer. It will perform an automatic selftest and proceed with booting itself. After some minutes you should see the

login

message.

- At last you have to connect the BBK-S4 link connectors via LNK-10 link adapter cables (normally 4 per link adapter) with an external transputer system (e.g. XPLORER or MULTICLUSTER-3).

Now you have finished the hardware installation successfully. The software installation of BTSS is described in chapter 2

4.2 S-Bus Specific Parts for Software Installation

To do the software installation process refer to chapter 2. That section will refer to this section to do S-Bus specific installation steps.

S-Bus specific installation steps are necessary because the BTSS needs read/write permission to the group vmebus for the S-Bus device. The procedure to set the correct premissions depends on the operating system:

4.2.1 Preparing S-Bus device under SunOS

- Give read/write-permission for the S-Bus device corresponding to the slot you have chosen for the BBK-S4. The S-Bus device is named

```
/dev/sbus<slot number>
```

First change the group membership to the vmebus group:

```
chgrp vmebus /dev/sbus<slot number>
```

To give read/write-permission to this device for the vmebus group, type the following command:

```
chmod g+rw /dev/sbus<slot number>
```

4.2.2 Preparing S-Bus device under Solaris

The handling of devices under SOLARIS has been changed considerably against Sun-OS: All generic device drivers have a *.conf* file, such as *sbusmem.conf*. One has to make sure that the file */kernel/drv/sbusmem.conf* exist, it contains entries for the various S-Bus slots.

- Edit the file */etc/minor_perm* to assign owner and group id's and appropriate permissions for the S-Bus memory device drivers. E.g. to permit read/write access to slot 2 memory for the group vmebus the file */etc/minor_perm* should contain something like:

```
...  
sbusmem:slot2 0660 root vmebus    # \BBKSiv\ in slot 2  
...
```

- Edit `/etc/devlink.tab` to tell where each driver is located (Note that fields must be separated by tab stops: `\t`):

```
...
type=ddi_pseudo;name=sbusmem sbus\A1
...
```

- Shutdown the system in the normal way.

Hint:

If the BBK-S4 has not been installed yet, this is a good moment to do. Powerdown the system and follow the hardware installation instruction in section 4.1

- Reboot the system with the following command:

```
> new [ENTER]
ok boot -r [ENTER] 1
```

- Login as superuser.
- Check that the links `/dev/sbus<slot number>` have been created. Set the permissions for the device corresponding to the chosen slot number to have read/write permissions for the group `vmebus`.

¹'>' and 'ok' are system prompts

Chapter 5

Vme-Bus Link Adapter

This chapter describes the VME-Bus specific installation steps. VME-Bus link adapter are BBK-V2, BBK-V4 and VMTM.

5.1 Hardware Installation for Vme-Bus Sun's

5.1.1 Link Adapter Overview

PARSYTEC provides the following VME-Bus link adapter for use inside SUN systems:

Board type	form-factor	bus master facility	software configurability	number of onboard transputers
BBK-V2	6U	yes	no	1
BBK-V4	6U	no	no	-
VMTM	6U	no	yes	4

The 6U link adapters can directly be used in SUN 6U VME-Bus slots; for using these link adapters inside SUN 9U VME-Bus slots a special form-factor adapter is required. PARSYTEC offers an adapter frame (SUN-FRAME) which already contains five lemosa link connectors for accessing external transputer networks via link cables.

Modules, that are able to act as bus masters on the VME-Bus requires special attention during their installation.

5.1.2 Hardware Requirements

In order to use VME-Bus link adapter, your SUN must have a free VME-Bus slot. Both form-factors, 6U and 9U, are possible (see 5.1.1).

5.1.3 Preparing the installation

This section is intended to be read before the actual installation takes place. It deals with designing the address-assignments for all the link adapters, jumper settings etc.

5.1.3.1 Planning the VME address-assignments

In order to plan the assignment of VME-Bus addresses for transputer modules, you must get well informed about the address usage by existing link adapters. For basic information refer to SUN's documentation "Writing

device drivers". It describes in detail the structure of all VME devices and the location of reserved areas. The command `dmesg` displays the booting diagnostic messages. This way you can figure out the VME addresses of existing devices.

For the default addresses and required area sizes of the transputer link adapters, see the following table:

Board type	default addr	size	remarks
BBK-V2	\$8000 0000	2 Mb	unused space (4Byte) required following. See note below
BBK-V4	\$8000 0000	256	
VMTM	\$8000 0000	256	

Note for BBK-V2 only:

The BBK-V2 has been designed as a multi-purpose VME-Bus link adapter. It offers the special feature of extending the local dual-ported memory by external VME-Bus memory, located at the contiguously following address. For example if you use the default address \$8000 0000 for BBK-V2 and add 4 Mb memory at \$8020 0000, the transputer will notice 6 Mb of contiguous local memory. This nice feature should be kept in mind when designing the address layout. The software development products determine the amount of local memory during boot-time by verified writes to various locations. If you have installed memory or used this next VME address at all, there may be a conflict in usage of it.

Best way to proceed would be to have a small (about 256 Byte) of unused VME address space following the last address used by a BBK-V2.

Note for Sun 4-110 or Sun 4-140 only:

Due to a scaled down VME-Bus interface, these SUNs lack the 32Bit (extended) addressing mode. Therefore it is necessary to do several changes

both in hardware- and software installation. All (with a single exception) default board addresses are 32 Bit addresses. Change them to 24 Bit addresses. SUN reserves addresses starting with \$80 0000 for OEM devices.

During the software installation the name of the VME-Bus device has to be changed from */dev/vme32d16* to */dev/vme24d16* inside the file *lconf.c*. But this is described in the software installation procedure.

5.1.3.2 Planning Sun slot usage

Plan the assignment of free SUN slots (6U or 9U) to the according link adapter.

Put special attention on the different form factors and the ability to become bus master (see 5.1.1). For every link adapter which may become a VME-Bus master, it is absolutely necessary to remove two jumpers at the backplane of the SUN. Due to the fact that link adapters do not become bus masters during normal operation, the position in the daisy chain is of minor interest. The last slot in the SUN is a very nice location for a single link adapter which is able to become bus master.

5.1.3.3 Setting the selected addresses

After having decided how to use the VME address space for all your link adapters, set the required address jumpers accordingly on all link adapters.

5.1.3.4 Checking the jumper settings

It is advised to check for all your link adapters the jumper settings. Each link adapter is carefully checked before delivery. This check includes the factory default jumper settings. But due to the human nature errors may occur. The following table lists the jumpers of special interest of each link adapter.

Note:

If you are already using PARSYTEC link adapters, please check that all link adapters have the same link-speed set. The former standard (some years ago) was 10 MBit/s, the new standard is 20 MBit/s.

All link adapters are delivered now with the 20 MBit/s link speed as factory default value.

Board type	Jumper	Description
BBK-V2	J 4	bus arbitration level 3
	J 5	all privileges, both data types
	J 8	32 address-bits, same as J 5
	J 33	link 0 to VME-Bus adapter
BBK-V4	none	
VMTM	J 7	connect transputer A link 0 to VME adapter 0 and link-switch config-link to transputer A link 1

5.1.3.5 Mounting 6U link adapters into the adapter frame

Only under rare conditions it is necessary to mount a link adapter into the form factor adapter. These conditions are:

- You are using a 6U VME-Bus link adapter and
- You plan to install it into a SUN 9U slot and
- You have not ordered PARSYTEC's form factor adapter (SUN-FRAME) together with the link adapter (in this case the link adapter will be delivered mounted into the frame to you)

The link adapter has two ejectors. One on the top, the other on the bottom of the front panel. Dismount both ejectors. Otherwise the link

adapter would not fit into the SUN-FRAME. Release the two mounting braces of the adapter frame and insert the link adapter into the frame. Pay special attention to firmly insert the connectors. Fix the board in the frame with the two mounting braces.

5.1.4 Performing the hardware installation

Hint:

Before proceeding with the hardware installation read carefully all the installation procedure, please. The software installation section should be read completely before you continue.

The installation of the hardware requires the computer to be powered down. Power down the system.

But be careful:

Hint:

If you have any doubts about knowing exactly how to power down the system, ask your system manager. The filesystem may be damaged if you do not obey the correct powerdown procedure!

- Power down your system. Be sure that no other users are logged in or they will be disturbed by your power down. This may happen if this system is used as a file server. Usually the command

```
shutdown -h
```

will work fine. After some time the message

```
unix halted
```

should appear. Otherwise contact your local system manager. Now you can switch off the power-supply of the computer. Remove the power cable from the computer for security reasons.

- Remove the front cover from all the selected SUN slots as planned in 5.1.3.2. Remove the metal dummy board inside those slots.

Important note:

If you are using a BBK-V2 link adapter you **must** remove the two jumpers described below (INT and BG3) anyway.

Though this seems necessary only when using these link adapters as bus masters or interrupt server/requesters their VME-Bus controller chip will get confused if these jumpers are present. You will be very surprised about the strange behaviour of your computer!

If you are using a VMTM or BBK-V4 the two jumpers must remain installed because those link adapters are pure passive bus components. They did not connect the daisy chains for bus-grant (BG0..3) or interrupt acknowledge (INT).

Important note:

The next two paragraphs apply to BBK-V2 only. If you are using a VMTM or BBK-V4 skip them.

- Locate the backplane jumpers for the VME-Bus. This is strictly machine dependend.

For more details see the SUN's documentation "Card Cage Assignment And Backplane Configuration Procedure".

Have a look at the backplane jumpers of the VME-Bus. For each slot you will see five possible jumper positions. In one group there are four jumpers near to each other and one separated by about one inch from the others.

Remove jumpers from two positions:

- The one standing apart from the group of three (INT).
- For BBK-V2 link adapter only:
From the group of four the one nearest to the lonesome jumper (BG3).
- Make sure that both daisy chains from slot 1 up to the selected slot are not broken anyway. This means that every slot between the chosen one and slot 1 has either the two jumpers installed or contains a board (maybe disk-controllers, memory extensions etc.).

From here on all boardtypes are concerned.

- Insert all boards or form factor adapters (SUN-FRAME) into the chosen SUN slots. Pay special attention to firmly depress the frame into the computer. The connectors must have good contact. Use the two screws on top and bottom of the board for this aim.
- As the final step reconnect the power cable and switch on the computer. It will perform an automatic selftest and proceed with booting itself. After some minutes you should see the login message. Remount the cover plates.

Now you have finished the hardware installation successfully.
The software installation of BTSS is described in chapter 2

5.2 Vme-Bus Specific Parts for Software Installation

To do the software installation process refer to chapter 2. That chapter will refer to this section to do VME-Bus specific installation steps.

5.2.1 Vme-Bus Specific Requirements

The VME-Bus device is named `/dev/vme32d16` (resp. `/dev/vme24d16` for 24 bit VME-Bus devices). Give read/write-permission ¹ to the group `vmebus` for the VME-Bus device. First change the group membership to the `vmebus` group. Type this command:

```
chgrp vmebus /dev/vme32d16
```

To give read/write-permission to this device for the `vmebus` group, type the following command:

```
chmod g+rw /dev/vme32d16
```

5.2.2 Vme-Bus Customization of BTSS

- If you are using a SUN 4-110 (or SUN 4-140 as well) it is necessary to change the default VME-Bus device from `/dev/vme32d16` into `/dev/vme24d16` due to the lack of 32 Bit addressing mode on a SUN 4-110.

During the installation you have to choose the link adapter type for 24-bit VME-Bus. Answer the question

Install for 24 bit VME-Bus? (y/n) :

by 'yes', but then the question for 32-bit VME-Bus you must answer by 'no'.

- Electronic Link Switch Configuration within the VMTM board:

The process of telling each link-switch, which connections should be established (the topology) is called 'configuration'. Do not mix it up with the configuration of software, which means basic adaption to local needs.

¹If you are working under Solaris, please contact Parsytec, how to give read/write permission to the VME-Bus device.

By default the topology for a VMTM board is described by the *vmtm.ncu* file. If none of the supplied sample topologies (*vmtm*.ncu*) fullfills your requirements, please refer to section 5.3 for details creating adapted topologies.

- After complete BTSS installation it seems very usefull to establish the chosen topology on VMTM link adapters during the power-up procedure of the SUN. The *ulti* command might be used for this aim. Include the following statement inside your */etc/rc.local* file:

```
ulti -c
```

This will silently configure and test all VME-Bus link adapter. Only error messages are written to *stdout/stderr*. You may redirect both outputs similar to other entries inside */etc/rc.local*.

5.3 Configuration Related Details

5.3.1 Description of the VMTM

The VMTM link adapters are fully electronically configurable boards. This means there is a NCU (network configuration unit) on board, which consists of electronic link-switches (INMOS C004).

The VMTM contains an onboard-NCU, four transputers with 1 or 4 MByte memory each, four links and nine external connectors. The connections of transputer A link 0 and the config-link of the link-switch are jumper selectable.

For more details please refer to the VMTM technical documentation.

5.3.2 Motivation for the Configuration

An above mentioned link-switch is an electronic cross-bar for 32 links. By writing simple command-bytes to an additional config-link, all the

desired interconnections are made (or removed). Two connections are jumper selectable. They exist independent of any power-supply.

As already stated, the link-adapters may be connected freely to any link on the board, so they have no connection at all after power-up, except the jumpered connections on a VMTM.

The process of telling each link-switch, which connections should be established is called 'configuration'. Do not mix it up with the configuration of software, which means basic adaptation to local needs.

Configuring the VMTM means establishing a fixed topology onboard. The topology will connect each usable link-adapter with a transputer link. This enables multiple users to use their (possible different) transputer-networks completely independent of each other.

The availability of those link-adapters are reported by a SUN utility program called 'links'.

Do not confuse transputer-links with link-adapters (gateways to SBus) and UNIX hard- or softlinks.

5.3.3 Description of the Configuring Process

Once the administrator in charge for the transputer-system has defined the required topology and installed it properly, it is established automatically at UNIX boot-time. Usually there is no need to configure manually.

The user interface to configure/test all available transputer boards in your SUN is the command named `ulti`. Call `ulti -h` to get a short usage information. The `-v` option stands for **verbose** and gives a lot of detailed information to you.

Without the `-c` option, all the available link-adapters are tested. This testing consists of opening the according link, resetting it, booting the transputer with a small checking program. Then a single byte is sent down this link, expecting it back incremented by one. The used link is closed afterwards. Without the verbose option, only errors are reported.

Supplying the configure option, each link is checked against being a config-link. If the link turns out to be a config-link, it will be configured.

An additional UNIX process is `vfork(2)`-ed and `exec(2)`-ed with the VMTM configuring program. Its default name is `/usr/etc/transp/vmtmconf`. This will boot the transputer A with the `vmtm.cde` binary. It is a simple bidirectional forwarder to link 1, which is the config-link for the link-switch. Forwarding means the link-adapter seems to be directly connected to the link-switch. This gives the advantage of very simple handling the link-switch directly connected to a link-adapter.

The topology as defined inside the included file `vmtm.ncu` is established by issuing several `connect` commands to the link-switch. It will be verified by sending the `status` command for every link to the link-switch. This will be displayed in verbose mode as vector of physical connections.

Regarding the vector as array and assume the 'k-th' index contains the value 'i', means, that the physical link 'k' is connected to physical link 'i'. This connection is bidirectional, so the index 'i' contains the value 'k' as well. See VMTM technical documentation for a map of physical to logical links (in directory `/usr/etc/transp/config` file `vmtm.h`).

By specifying the '-r' (read) option to `ulti`, no configuration is done, but the existing topology is read and displayed as described above.

Finally the `vmtmconf` program will terminate, reporting success or failure to configure through the return value.

Note:

`vmtmconf` is not intended to be called directly by a user. Instead it is executed by `ulti`, the user interface program for testing transputers and configuring them.

In order to configure for the first time give the command

```
ulti -cv,
```

followed by using

links

for a report of link-adapter availability.

Since the `ulti -cv` command may be called any time, it is possible to reconfigure the board while another application program is still running on several transputers on the VMTM. During the time needed for configuration all links are disconnected which may lead to loss of data on the links. So the usage of `ulti -cv` has to be considered carefully after the initial configuration.

The `ulti` command may be called from any current directory. The `ulti` program must know where to find the configuring `vmtmconf` program. There is a default pathname `/usr/etc/transp/vmtmconf` for that, but it may be overridden by the environment variable `'TRANSP_VMTM_CONF'`.

The `/usr/etc/transp/config` directory contains all the supplied programs in source form, together with all needed include files. The desired topology is described in the file `vmtm.ncu`. The `makefile` offers you a variable to change this. The configuration file is included into the C-program, so you can use all C-style comments and macro processing power. Now the `vmtmconf` program must be recompiled. Simply call `make` to do so. Configuring this new topology requires one point to be kept in mind:

- the default location `/usr/etc/transp/vmtmconf` may be overridden by environment variable `TRANSP_VMTM_CONF`

5.3.4 Definition of the Topology

The topology information for a VMTM board is contained entirely inside the file `vmtm.ncu`.

There are two different example topologies, called `vmtm.ext.ncu` and `vmtm.std.n`. The file `vmtm.ncu` is a copy of `vmtm.ext.ncu`.

It is intended to be used as connection to external transputer systems like a MultiCluster 2 or SuperCluster. The topology is a 'zero-two' pipe with a single transputer supporting four independent users.

The former standard topology is contained in *vmtm_std.ncu*. It is a cyclic 'two-three' pipe over all four transputers with their link zero connected to the VME-Bus. So a single user may easily boot a four transputer network.

If you plan to create a topology on your own, you may start with one of the existing files applying several modifications to it. The following information may be helpful for you.

The *vmtm.ncu* file will be included into *vmtmconf.c*. The file *vmtm.h* 'defines' the symbolic names for links on the VMTM board.

If you make any mistakes in specifying the topology, they can not be detected during compile-time. At run-time there is a check of the requested topology (e.g. two links connected to single third link) and error messages will be passed to you.

Chapter 6

Parsytec Standard Installation

The PARSYTEC Standard Installation is a product independent tool to install software products in a standard way. Usually it is delivered together with PARSYTEC products.

The PARSYTEC Standard Installation searches for all `.tar` files in the actual directory and offers them for installation. A file with the same base name and the extension `.descr` may contain a product description and targets of a product specific Makefile. These targets may be done during the installation process. The product specific Makefile is located inside the `.tar` file in the product's directory `install`.

The PARSYTEC Standard Installation runs under SUNOS 4 and SOLARIS (if the product specific Makefile does).

Experienced users may **start the product installation** by reading the next section (6.1 Fast Installation Guide). For other users or in case of problems a detailed installation guide is available after the next section.

6.1 Fast Installation Guide

For installation requirements and product specific hints introduce the product's documentation. Please check the product specific installation requirements (if available) before you start the product installation.

6.1.1 Basic Installation

A product installation may be done by a few steps:

1. **Make a temporary directory** on a disc with enough space left and change to this directory.
2. **Read the product tape** (floppy disc, ...) to the temporary directory:

```
tar xvpf /dev/...
```

3. Start the installation process (as superuser) by:

```
make install
```

4. **Answer the online questions** and read the text carefully. (cf section 6.2.7)

For installing the same product on another workstation login there and repeat the process. After all desired products have been installed remove the temporary directory.

6.1.2 Reinstallation

Some products may need one (or more) **reinstallation** step(s) (e.g. to generate workstation specific servers). To do this repeat the installation process in the root directory of the installed product:

1. Change to the product's root directory.
2. Start the reinstallation process (as superuser) by:

```
make install
```

3. Answer the online questions and read the text carefully. (cf section 6.2.7)

After starting the installation process you will get a list of all installed products which are now indicated by a '+' (instead of a '*').

6.2 Detailed Installation Guide

This chapter describes in detail the product installation by the PARSYTEC Standard Installation. Further information on customization etc. can be found in the following sections.

The installation steps are:

- Check the installation requirements (cf section 6.2.1 and product documentation (installation chapter))
- Read the tape (cf section 6.2.2)
- Control the delivery (cf section 6.2.3)
- Decide which products you want to install
- Search a location for each product and create a directory (cf section 6.2.4)
- Do the product specific preparations (cf product documentation (installation chapter) and section 6.2.4)
- Start the installation process (cf section 6.2.5)

- Answer product specific questions (cf section 6.2.7 and product documentation (installation chapter))
- Do the complete installation (cf sections 6.2.5, 6.2.6 and 6.2.7)

6.2.1 Requirements for Installation

Following things are required to use the PARSYTEC Standard Installation:

1. A SUN4 running SUNOS 4 or SOLARIS
2. A suitable device for reading the tape, floppy disc, etc. PARSYTEC products usually are delivered on a 1/4" streamer tape readable on a QIC-150 tape device. Normally the tape contains all software products you ordered.
3. We assume that you are using `cs`h or `tc`sh as a shell. If you are using a different shell, the commands might be slightly different.
4. Enough amount of memory to read the tape to a temporary location.
5. Enough amount of memory on the disc where you want to install the product. Use the command `df` to check the free disc space. (After reading the tape you may get the product space requirement by looking at the `.tar` files by `ls -l`.)
6. superuser permission (in special circumstances you may try it without being superuser, but you may fail on insufficient permissions to change owner and group or on insufficient write permission to directories of already installed products).
7. A `make` in your path, call `which make` (GNU `make` may not work in any case).

Note: if you have installed SOLARIS you may get the message `make: Command not found`. This may happen because there is no `/bin/make` for the normal installation of SOLARIS. You should add `/usr/ccs/bin` to your path variable typing

```
set path=($path /usr/ccs/bin)
```

Further requirements on hardware, software products, etc. are product specific and are listed in the product documentation, chapter Installation or in the online product information (cf section 6.2.7). Dependencies on other software products usually are defined in the product's .descr file. The dependency then will be controlled during the installation process.

6.2.2 Read the Tape

1. Login as the superuser.

```
login: root
```

```
password:
```

or

```
% su root
```

```
password:
```

2. Search a temporary location for the tape contents:

For the tape contents, locate a disc partition containing enough free space, create a temporary directory (let's assume this directory is named tempdir):

```
mkdir tempdir
```

and change the current directory to this location:

```
cd tempdir
```

This directory will only be used to read in the tar files from the tape and to start the installation.

The temporary directory has to be empty and must be different from the directory where you actually want to install one of the products.

3. The software usually is distributed on a 1/4" streamer tape readable on a QIC-150 tape device. Insert that tape into your tape device usually named /dev/rst0 (for SUNOS) or /dev/rmt/0 (for SOLARIS). The tape rewinds automatically. You may list the tape contents for SUNOS:

```
tar tvf /dev/rst0
```

or for SOLARIS:

```
tar tvf /dev/rmt/0
```

4. In order to read the tape, supply the following command for SUNOS:

```
tar xvpf /dev/rst0
```

or for SOLARIS:

```
tar xvpf /dev/rmt/0
```

This copies a set of tar files (one for each software product) and the files necessary for the installation to your disc. The directories where the product trees will be installed can be specified later on. They must not be the same directory where you read the tape contents.

6.2.3 Checking the Delivery

Within any software delivery a set of installation tools is delivered on the same tape. Each software product on the tape consists of at least one

tar file (extension `.tar`) and — in most cases — of an additional product description file (extension `.descr`).

The delivery tape may contain one or more products.

To check your delivery, login as superuser¹, change to the temporary directory where you read in the tape. Call `make install` and wait until the menu

```
Available software components for installation :
```

appears. After this line you will get one line for each available product containing the product number 'no.', the 'state' ('*' = available for new installation, '+' available for reinstallation), the 'product' name and its 'version'. Please check if all products you ordered are available.

By typing '? <product number>' you can get a short product description and informations about installation requirements for the product ² (online documentation).

Please study the requirements carefully while reading the next section. Do all the preparations listed in the following section, in the online documentation and in the product documentation.

You can interrupt the delivery check by typing 'q' for quit³.

6.2.4 Product Specific Preparations

During the installation process a number of product specific questions are asked. The questions cover the product itself and products of interdependency. It may depend on products which will be installed within the same installation process or on products already installed some times ago.

¹You need not be superuser to check the tape contents, but note that a warning will be printed after starting `make install`

²This works only if an associated `.descr` file exists.

³Interrupting the installation process by 'q' the surrounding makefile terminates with an error message. Please ignore it.

The interdependency between all products of the same installation process is defined via the products' .descr files. So the correct installation order is found automatically by the PARSYTEC Standard Installation.

To prepare the installation process consider the following items:

1. Decide which products you want to install.
2. **Check the installation requirements** for each product. The requirements may be:
 - special hardware
 - a special operating system, a special version or special options to be installed
 - special software, which may be needed but was not delivered and is not installed yet (install it before!).
Check also the version of required software.

Locations where you may find the product specific requirements are:

- the product documentation (installation requirements)
 - the PARSYTEC Standard Installation online documentation (cf section 6.2.3 and 6.2.7)
3. During the installation, you are asked for each product for the **path where you want to install this product**. Please decide it for every product after you checked the requirements. Create the directories if they not yet exist. For new product releases the directory should be empty.
 4. Decide which **owner and group** the product should belong to. The owner and group you enter must exist, otherwise they are not accepted. Please create the desired owners and groups before installation. Please use the user and group names as recommended, as these names may be used throughout the whole documentation.
 5. Decide for each product if **access rights for others** should be given. They can be removed during the installation.

6. Some products are dependent on other, already installed products. Find out **where these products are located**.
7. Think about **installation in a workstation network**. Do you want to install the product in a global location (mount the discs), or do you want to install the product local on several workstations. In the second case you have to repeat the installation on every other workstation for the local products.
8. Check which reinstallations are needed. Reinstallation steps may be necessary e.g. for workstation or operating system specific servers or after changing a configuration file to your special configuration so that configuration specific installation may be done afterwards in a reinstallation step (cf section 6.1.2 or 6.2.8 and online hints).
9. Do other preparations given in the product documentation (installation chapter) or in the online documentation.

6.2.5 Starting the Parsytec Standard Installation

After reading the tape contents (cf section 6.2.2) you may start the installation process:

1. **Login as the superuser:**
superuser permissions are required during installation for changing owners, groups and permissions of files and probably to write to other product installations.

```
su root
```

2. Change to the temporary directory where you read in the tape contents:

```
cd tempdir
```

3. In order to install all desired products, type

```
make install
```

`make` (do not use GNU `make`) first starts a program which asks questions about the system configuration. Afterwards a makefile is started which does the product installations.

All products on the delivered tape may be installed in one single step. For an installation on a workstation network, some installation steps have to be done on other workstations as well. To install the complete product on another workstation repeat the above process on that workstation.

6.2.6 Installation Program

The PARSYTEC Standard Installation is started by typing `make install` in the temporary directory containing the `.tar` files (or in the product's root directory in case of reinstallation).

The installation program prints the available products, asks questions for each product (cf section 6.2.7) and generates a makefile named `install.mk` containing all product and system specific information. This makefile is executed and controls the installation process by accessing targets in the product specific Makefiles. System and product information are passed to the product specific makefile via a file with the extension `.dat`.

The installation terminates by printing:

```
* installation finished *
```

6.2.7 User Interface

During the installation two types of questions are asked: the first type can be answered by 'y' (yes) or 'n' (no), default is 'y'. The second type requires text input. The default settings shown in brackets can be chosen

by hitting <RETURN>. To stop the installation process any question may be answered by 'q' (quit).

Having started the installation process you will get a list of all available products (i.e. .tar files in the temporary directory) which may be installed. They are listed in the menu

Available software components for installation :

To get product specific information (online documentation) enter '?' followed by the *product number*. You will get the following informations:

- product description
- installation requirements concerning hardware and other software products
- list of commands which must be available during the installation

Choose the products desired for installation by typing the *product numbers* separated by ','. (The order of installation is not dependent on your input but on the product dependencies.)

A list with your chosen products will appear and you are requested to confirm this list.

After accepting the product list, your workstation system is checked for some requirements:

- the workstation operating system,
- the workstation operating system release,
- the workstation machine (hardware) type,
- the workstation processor type

If your system has a component which is not recommended you will see a warning, but you can continue the installation if you want to.

For every product to be installed a set of **standard questions** are asked, other questions concern product specific things. The standard questions are:

- Enter the (full) path name for '*product*'
Where do you want to install it ?

Enter the location where you want to install the product (you may use '~' for a user's home directory). If the directory doesn't exist you may create it. You need write permission for the directory.

- Enter the owner for '*product*'

All product files and directories (that need no special ownership) will be changed to the owner you will enter. You need the permission to change to the owner you enter.

- Enter the group for '*product*'

All product files and directories (that need no special group) will be changed to the group you will enter. You need the permission to change to the group you enter.

- Give access rights to others for *product* ? (y/n)

If you enter 'yes' the permissions for others will not be changed (usually permissions for others are set). If you enter 'no' all other permissions (rwx) will be reset for all files and directories (exceptions may occur).

For all standard questions product specific hints may occur before the question.

The installation process will go on asking **product specific questions**. Please refer to the product's documentation (installation chapter). Product specific questions may concern product interdependencies which are checked by the PARSYTEC Standard Installation. Dependencies between

the products which are actually going to be installed will be solved automatically. You may confirm the proposed product or not.

If no suitable product is found you have to enter the path where a product is installed.

If a product has been located by the PARSYTEC Standard Installation it suggests a product in the following form:

The installation of '*product*', target '*target*', may use the following '*category*' software:

<i>name of the product</i>	
directory	: <i>product path</i>
type	: <i>product type</i>
version	: <i>product version</i>

Use it ? (y/n) :

category is a keyword to identify products. Installed products with another *category* are not suggested.

product path is the directory where the product will be installed (installation in the same step) or where the product is installed (installation already done before).

product type may either be the type of the product inside the product category, or it is uninstalled or unknown. uninstalled means that the directory not yet exists, unknown means that the product type cannot be found out (A reason may be that the product was not installed by the PARSYTEC Standard Installation.).

product version is the version found out by the PARSYTEC Standard Installation or it is unknown. unknown means the product's version is not available within the product installation. A version control cannot be done automatically so you have to ensure manually that the required version is installed.

If type and version are not unknown the PARSYTEC Standard Installation checks the product for the required version and required files. If a mismatch is found a warning will be printed:

Warning: the installed software ...
does not fulfill the file requirements ...

or:

does not fulfill the version requirements ...

We strongly recommend not to use old or incomplete software, therefore answer the question:

Use it anyway ? (y/n) :

by 'no' to choose another installation or stop the installation process by 'q' (quit) and update the concerned software before installing the actual software.

After the questioning has finished for one product you may reanswer the questions if you type 'y' for the following question:

Do you want to CHANGE YOUR INPUT for ...

After the questions for all products have been answered, the installation proceeds with:

Would you like to do the above parts of the installation ?

If you answered all questions correctly, you can start the installation process by 'y' (yes). If you type 'n' (no), the installation is not done (although the product directories may already have been created) and you may start from beginning without removing any done installation.

Starting the installation it starts to unpack and install all products ⁴. The installation prints a succeeded message for every product.

After this, the basic installation on your system is finished. If you want to install the same software on other workstations repeat the installation procedure.

Don't forget to remove the temporary directory, where you read in the tape.

⁴From the configuration information a makefile `install.mk` is generated and started. This makefile copies the product files to the specified locations, calls product specific targets and changes owner, group and permissions as desired.

6.2.8 Reinstallation

After the basic installation was done some products may need one (or more) reinstallation step(s) (e.g. to generate workstation specific servers). To do this repeat the installation process in the root directory of the installed product.

Suppose a product (e.g. PARIX) needs a reinstallation for a workstation specific server (e.g. named dserver) on your second workstation (e.g. named pluto). Let's assume, the product is installed in the path /parix then following steps are done for a reinstallation:

1. login on your second workstation:

```
rlogin pluto
```

2. become superuser there:

```
su root
```

3. Change to the product's root directory (e.g. /parix):

```
cd /parix
```

4. Start the reinstallation process by:

```
make install
```

5. Answer the online questions and read the text carefully. (cf section 6.2.7)

After starting the installation process you will get a list of all installed products which are now indicated by a '+' (instead of a '*').

The PARSYTEC Standard Installation behaves like in the basic installation, some questions may be modified or may disappear.

To guess which reinstallation steps are necessary for a product introduce the product documentation (installation chapter) and the PARSYTEC Standard Installation online documentation (cf section 6.2.3 and 6.2.7).

6.2.9 Installation Stuff inside the Product Directory

Installing a product the PARSYTEC Standard Installation copies itself and some relevant installation file to the product directory.

The following files are copied to the product's root directory:

- `Makefile` : global makefile for installation
- `p_config` and `p_config.SunOSx` : installation control program (for SUNOS 4 and SOLARIS)
- `p_config.rsc` : resource file for `p_config`
- temporary files may be : `install.mk`

Some files are stored in the `install` directory:

- `product_base_name.descr` : Product Description File (May be recovered by a reinstallation or a future installation done by the PARSYTEC Standard Installation. It will eg be used to get the product's version number)
- `product_base_name.make` : Product Specific Makefile(May be recovered by a reinstallation done by the PARSYTEC Standard Installation.)
- temporary files may be : `product_base_name.dat`, `Descrfile`, `Makefile`

6.3 Installations on Additional Workstations

If your workstations are connected via NFS, it is possible to install products on additional workstations without installing the product directory tree on the workstations.

Some products are usable without further steps others need a reinstallation step (eg workstation specific servers have to be created for PARIX). The third kind of product needs a complete new installation if the other workstation drives another operating system (eg MOTOROLA compilers).

You may install a product for another workstation without login on that workstation if the superuser of the actual workstation has access to the target directory. But note that reinstallations sometimes must be done on the target workstation itself.

In order to install or reinstall a product on the additional workstation, first login there:

```
rlogin NewWorkstation
```

Afterwards refer to the sections 6.2.5 ff, for reinstallation refer to section 6.2.8.

6.4 Installing a Patch Set

After some Software Problem Reports (SPR's) have arrived, products will be improved between the major releases. Sometimes it may be necessary to change the actual product installation. In this case the patch will be distributed by floppy disc, tape or email.

The patch can be installed in exactly the same way as a new product (cf 6.1 or 6.2). When asked for the path of the product tree, you should give the location of the already installed product.

A patch may be installed together with other products.

Please read the online documentation (cf section 6.2.3 and 6.2.7) for the patch carefully. If you eg install both, the product and the patch, some steps for the product may be skipped and be done in the patch installation.

6.5 Difficulties

If you have any difficulties installing PARSYTEC products, you should first reread the documentation (PARSYTEC Standard Installation, PARSYTEC Standard Installation online documentation and the installation chapter of the product specific documentation). If this does not help, contact your distributor or PARSYTEC by email or phone. Our coordinates are:

email: `support@parsytec.de`
phone: +49 241 8889-0 (ask for the support people)
fax: +49 241 8889-50 (indicate support)

When the PARSYTEC product PARIX is installed, the preferred method of submitting problem reports to PARSYTEC is to use the software problem report (SPR) system which is part of PARIX. This tool will help you to collect a complete description of the problem, and supports us in tracking it down. Its customization and its use are described in the PARIX documentation.

Chapter 7

BTSS Tools

The BTSS includes two tools for testing and diagnosing link adapters: `links` and `ulti`. Furthermore, some bus specific tools exist (eg `burn_scsi` allows you to install new firmware versions into the BBK-SCSI). The commands are described in the following sections.

7.1 links

The command `links` may be used to print status information about all hardware links to transputers and may result into the following output:

Display status about all hardware links to transputers

Id	Status	Link-Name	Device
0	available	scsi-t b0 10	/dev/rmt/0
1	available	scsi-t b0 11	/dev/rmt/1
2	IN USE	scsi-t b0 12	/dev/rmt/2
3	available	scsi-t b0 13	/dev/rmt/3

In this example you get the information, that one BBK-SCSI with four links is available and that link 2 is currently in use.

7.2 ulti

The command ulti may be used to test the link hardware and the connected transputers of a link adapter:

```
usage: ulti [-v] [-c] [-r] [-t test-program] [-l link-name]
        configure (set topology) and test accessibility of Parsytec transputer
        -v verbose output
        -c configure all boards (VMTM, MTM-SUN) with default topology
        -r read existing configuration and display it
        -t specifies other test-program (transputer binary)
        -l configure/test only this link and not all
```

If you call eg ulti -v you will get the output:

Hi there! Welcome to the ultimate transputer board test-utility.

The config file tells me the following configuration is available on your workstation.

Id	Status	Link-Name	Device
0	available	scsi-t b0 10	/dev/rmt/0
1	available	scsi-t b0 11	/dev/rmt/1
2	available	scsi-t b0 12	/dev/rmt/2
3	available	scsi-t b0 13	/dev/rmt/3

number of used links: 0

```
Testing link "0" ... OK
Testing link "1" ... OK
Testing link "2" ... OK
Testing link "3" ... OK
```

Have tested 4 of 4 transputers successfully

That means, one BBK-SCSI with four links is installed, it works correctly and all four links are connected to a 'living' transputer.

If one link is currently in use (eg link0), the `ulti` output for this link is:

```
Transputer link "0" currently in use
```

If one link is not connected to a transputer or the transputer is not switched on, you will get the output:

```
Testing link "0" ... FAILED!
  Link not connected to a transputer ( board not configured )
```

7.3 Tools for BBK-SCSI

These commands are available only if SCSI-Bus installation was done.

7.3.1 `getversion`

The command `getversion` is used to get information about the current used firmware inside the BBK-SCSI. It is located in the subdirectory `scsi_tools/SunOSx` of the BTSS root directory (x is 4 for SUNOS 4 and 5 for SOLARIS).

`getversion` takes the devicename assigned to one of the links of the BBK-SCSI as an argument. Usually this is the devicename entered into the `trans_conf` configuration file. You can use it as follows:

```
getversion /dev/rmt/4
```

The output should look as follows (you may get different version numbers and other information):

```
@(#) 1.13 Module: scsi, Part: SCSI-bridge.  
@(#) Host: SunOS aphrodite 4.1.3 7 sun4c, Target: T425 SCSI-T bridge  
@(#) Date: Tue Nov 22 09:38:12 1994
```

7.3.2 burn_scsi

The tool `burn.scsi` may be used to install a new BBK-SCSI firmware version. It is located in the subdirectory `scsi_tools/SunOSx` of the BTSS root directory (`x` is 4 for SUNOS 4 and 5 for SOLARIS).

The firmware of the BBK-SCSI is contained in a reprogrammable Flash-PROM. Therefore an update of it is easy possible without any hardware changes. The program `burn.scsi`, taking a special formatted file with a valid software version and the name of a device node connected to the BBK-SCSI as arguments, realizes the installation of a new software version. It reports possible errors during the process of burning the software. The Flash-PROMs are guaranteed to process a reburning 1000 times. If any error occurs during the phase of reburning you need to send in your BBK-SCSI to our support team for replacing the finished Flash-PROMS. A possible example for using `burn.scsi` is:

```
burn.scsi scsi_firmware_v1.5 /dev/rmt/0
```

The change of the firmware takes effect first after turning the BBK-SCSI off and on.

7.4 Tools for BBK-S4

These commands are available only if S-Bus installation was done.

7.4.1 s4diag

The command `s4diag` is used as diagnostics tool for the BBK-S4:

```
usage: s4diag [-h] [-i] [-r <board-no>] [-f]
           : without any argument display SBus slots
-h : help
-i : interactive mode
-r : reset board <board-no>
-f : force board reset (not interactive)
```

7.5 Tools for Vme-Bus

These commands are available only if VME-Bus installation was done.

7.5.1 vmtmconf

The command `vmtmconf` is used to configure the VMTM link adapter:

```
usage: vmtmconf [-v | -r] link-name
-v  verbose
-r  do not configure, display existing topology
```

For its use refer to section 5.3.3.

Note:

`vmtmconf` is not intended to be called directly by a user. Instead it is executed by `ulti`, the user interface program for testing transputers and configuring them.

Chapter 8

Public Link Interface

This interface provides you with all necessary functions to use the link adapter in your own applications with the streammode or with the packemode.

The functionality is similar to a device driver, but the implementation are two object files `link.o` and `lconf.o` located in the BTSS root directory. These must be linked together with your application.

The procedures offered by the link interface completely fulfill the specifications of the definitions given by Inmos together with some useful extensions.

The functions listed below are available in the object file `link.o` inside the BTSS root directory:

<code>int OpenLink();</code>	Open the specified link returning a <code>LinkId</code>
<code>int ReadLink();</code>	Read a number of bytes from the link <code>LinkId</code>
<code>int WriteLink();</code>	Write bytes to link <code>LinkId</code>
<code>int CloseLink();</code>	Close down the link <code>LinkId</code>
<code>int ResetLink();</code>	Cause a reset on the <code>LinkId</code>
<code>int TestRead();</code>	Check for input ready on <code>LinkId</code>
<code>int TestWrite();</code>	Check <code>LinkId</code> whether ready for output
<code>int InfoLink();</code>	Obtain informations about links
<code>int PacketMode();</code>	Switch the specified link to packet mode
<code>int StreamMode();</code>	Switch back to the default stream operation mode
<code>int GetPacket();</code>	Read one packet from the <code>PacketLink</code> into the buffer
<code>int PutPackets();</code>	Write as much packets to the <code>PacketLink</code> as possible

The file `parsylink.o`, which may be used instead of `link.o` contains the same functions with the prefix `Parsy` in the function names (e.g. `OpenLink()` is called `ParsyOpenLink()` in `parsylink.o`).

The file `lconf.o` contains BTSS configuration details, eg the BTSS default path, which are used in the above functions.

8.1 Internal Information

The access to the S-Bus or VME-Bus is done through the `mmap(2)` system call. A single page is mapped from the bus directly into the process virtual address space. This is very safe and fast as well.

Due to the byte-synchronous nature of a link there is no hardware buffer at the end of a link. The interrupt response time of a UNIX machine is not designed to be short. So the link adapters are driven in polling mode. Using an adapted polling strategy does not waste cpu-time and gives high performance.

For security reasons a group `vmebus` is used in conjunction with the 'set-gid' bit. Outside the interfacing functions the effective gid is always set to the real-gid.

A configuration text file (by default `/usr/etc/transp/trans.conf`) is used to describe the link adapters and their specification (SCSI-Bus: driver name, S-Bus: slot number, VME-Bus: address on VME-Bus).

A set of semaphores is used to guarantee exclusive access to each link. The semaphores used by a process are returned automatically on termination of the process even if it is killed by signal SIGKILL.

Development products, like PARIX, TOOLSET or PowerTools, access the link adapter hardware through this link interface. This is the reason for completely independent usage of all available links by different users and development products. User written applications share this feature as well when using the link interface. Additionally, a single process may open several links at the same time and operate independent from each other.

A detailed description of each link-interface function may be found inside `link_c/link.doc` of the BTSS installation directory.

There are some parameters which are not likely to be changed. In order to give maximum flexibility they are defined inside the file `link_c/lconf.c`.

8.2 Link Interface Function Description

A detailed description of each link interface function may be found inside the file `link_c/link.doc`:

Description of all the link-interface functions.

```
/* sccs id: @(#)link.doc 1.3 95/05/31 */  
  
/*  
 * OpenLink  
 *  
 * Open the link associated with 'Name'.  
 * If 'Name' is NULL or "" then any free link can be used.  
 * Returns any positive (including 0) integer as a LinkId or  
 * a negative value if the open fails.  
 *  
 * The syntax of 'Name' is:  
 *
```

```

* [<board-type> [<board-number>] [<link-adapter-number>] ]
*
* <board-type>  "BBKV2"|"BBKV4"|"BBKS4"|"VMTM"|
*              "SCSI-T" | <0..MAXSHORT>
* <board-number> B<0..MAXSHORT>
* <link-adapter-number> L<0..8>
*
*
* Besides the board-types listed above, a wide range of different
* spellings will be accepted as well. Usually small and capital
* letters are accepted (but not mixed up in a single word).
* Additionally '-' and '_' will be regarded as valid separators
* within a name. Also small letters for board and link speci-
* fications are ok. See the complete list of accepted names below.
*
* First a systemwide known configuration-file (trans_conf) is
* read into local memory. This file contains information
* about the type, number and VME-bus addresses of transputer
* moduls connected to this maschine. Each module is assigned a
* unique so called board-number to identify this particular
* board. The configuration file is formatted as follows:
*
* <board-number> <board-type> <device>
*
*
* where <board-number> and <board-type> are defined as above
* and
*
* <device> := <scsi tape device> | <sbus slot number> |
*           <vme-base-address>
*
* <scsi tape device> Tape device to be used for BBK-SCSI
* connection (eg.:
* /dev/rst0 : device for SunOS 4, uses rst0
*   to rst3
* /dev/rmt/0: device for Solaris, uses ../0
*   to ../3
* The entry needs to correlate with the device
* node for the first logical unit of the
* target which represents the BBK-SCSI.
*
* <sbus slot number>      A hexadecimal number according to
* C-style "%x" format. It specifies
* the number of the SBus slot used
* by this SBus device board.
*
* <vme-base-address> C style hexadecimal number "%x" format
* and gives the selected base address
* for VMTM e.g. "80200000"
*
* Each link will be locked after beeing opened by a system
* semaphore. So each link MUST be freed by calling 'close()'
* after using it. Otherwise nobody else can use this link.

```

```

*
* Complete list of accepted board-type specifications:
*
* bbkv2 bbk-v2 bbk_v2 BBKV2 BBK-V2 BBK_V2
* vmtm VMTM
* bbkv4 bbk-v4 bbk_v4 BBKV4 BBK-V4 BBK_V4
* bbks4 bbk-s4 bbk_s4 BBKS4 BBK-S4 BBK_S4
* SCSI-T scsi-t
*/

int OpenLink (Name)
char *Name;

/*
* CloseLink
*
* Close the active link 'LinkId'. Returns 1 on success or a negative
* error code on failure.
* It is an error to close a link which is not in open-state. The
* link is unlocked and unmapped.
*
*/

int CloseLink (LinkId)
int LinkId;

/*
* ReadLink
*
* Read 'Count' bytes into 'Buffer' from the specified link.
* 'LinkId' is a valid link identifier, opened with 'OpenLink()'.
* 'Timeout' is a non negative integer representing tenth of seconds.
* A 'Timeout' value of zero is an infinite timeout.
* The timeout is for the complete operation.
* If timeout is positive then ReadLink may return having read
* less than the number of bytes asked for.
* Return the number of bytes placed into 'Buffer' (which may be zero)
* or negative to indicate an error.
*
*/

int ReadLink (LinkId, Buffer, Count, Timeout)
int      LinkId;
byte     *Buffer;
unsigned int Count;
int      Timeout;

/*
* WriteLink

```

```

*
* Write 'Count' bytes from 'Buffer' to the specified link.
* 'LinkId' is a valid link identifier, opened with 'OpenLink()'.
* 'Timeout' is a non negative integer representing tenth of seconds.
* A 'Timeout' value of zero is an infinite timeout.
* The timeout is for the complete operation.
* If timeout is positive then WriteLink may return having written
* less than the number of bytes asked for.
* Return the number of bytes actually written (which may be zero)
* or negative to indicate an error.
* Actually written means send to transputer with acknowledge
* received.
*
* Implementation Note
*
* The used link-adaptor is a C011/C012. The output-status is
* assumed to be 1 allways. When sending a byte it is written
* to the output-data register and then the output-status
* register is checked (with timeout) until output-status is
* 1 again (1 means: ready for next byte).
* Once when entering this function the output-status register
* is checked for containing a 1. If this check failes, 'WriteLink'
* immedeatly returns 0.
*
*/

```

```

int WriteLink (LinkId, Buffer, Count, Timeout)
int      LinkId;
byte     *Buffer;
unsigned int Count;
int      Timeout;

```

```

/*
* ResetLink
*
* Reset the specified link line.
* 'LinkId' is a valid link identifier, opened with 'OpenLink()'.
* Returns 1 if the reset is successful, a negative error code
* otherwise.
*
*/

```

```

int ResetLink (LinkId)
int LinkId;

```

```

/*
* AnalyseLink
*
* Inmos says: Analyse the subsystem associated with the specified link.
* Parsytec boards have the famous feature of beeing individual reset.
* This reset always do the reset/analyse sequence.

```

```
* In order to offer all interface routines, we do a 'ResetLink' here.  
* Return values see 'ResetLink'.  
*  
*/
```

```
int AnalyseLink (LinkId)  
int LinkId;
```

```
/*  
* TestError  
*  
* Parsytec boards do not have any possibility to check the error  
* status of a neighbour transputer.  
* In order to offer all interface routines, we return always  
* zero, which means: No error set.  
* Inmos: Returns 1 if error is set, 0 if it is not and a negative  
* error code otherwise.  
*  
*/
```

```
int TestError (LinkId)  
int LinkId;
```

```
/*  
* TestRead  
*  
* Test input status of the link.  
* Returns 1 if ReadLink will return one byte without timeout,  
* 0 if it may not and a negative error code.  
*  
*/
```

```
int TestRead (LinkId)  
int LinkId;
```

```
/*  
* TestWrite  
*  
* Test output status of the link.  
* Returns 1 if WriteLink can write one byte without timeout,  
* 0 if it may not and a negative error code.  
*  
*/
```

```
int TestWrite (LinkId)  
int LinkId;
```

```
/*
```

```

* InfoLink
*
* Local extension done by Parsytec. This function gives
* information about all the link hardware on this machine.
* It is passed the address of a pointer to the 'ext_link_info'
* structure. If this pointer is NULL, enough memory is allocated
* and the pointer is changed to point onto it. Then the array is
* filled with information. Here a short description:
*
* link_number is the index in this array and the
* corresponding LinkId of this link.
* in_use 0 if link not in use, >0 if currently
* used (as recorded by the semaphore)
* or <0 if semaphore access has failed
* config_type If > 0 this link is a configuration link.
* The value describes the board type.
* A VMTM has the build-in ncu.
* See file 'link.h' for more details.
* The table lists the flagged link:
*   board link
*   =====
*   VMTM LO
* link_name is the link name as string, formatted:
* "nnnnnnnn bxx lx"
* n board name   (%-8s)
* bxx board number (%-2d)
* lx link number on board
* (%1d)
* link_address the VME-bus address of this link
* slot_no the SBus slot number (0..3) or -1 otherwise
*         tape device The tape device to be used for BBK-SCSI
*/

int InfoLink (LinkPtr)
struct ext_link_info **LinkPtr;

/*
* lerror
*
* Return a pointer to a string giving the explanation of
* the error-code supplied as a parameter.
* Usually error-codes are negative values. The positive
* error code value will be accepted as well.
* In case of an internal error (error-code out of range)
* the returned pointer points to an according error message.
*
*/

char *lerror (err_code)
int err_code;

```

```

/*
 * PacketMode
 *
 * Switch the specified link to packet mode. This link is called
 * 'PacketLink'. The link 'LinkId' must already have been opened
 * successfully by use of 'OpenLink()'. The specified link must
 * belong to a board supporting packet protocol (BBK-S4 only).
 * 'PacketSize' is the maximum size of any packet transferred over
 * this PacketLink. It is used by the firmware to make optimal
 * use of buffer memory on the BBK-S4.
 *
 * For a single user there is at most one link switched to packet
 * mode, which must not be used in stream mode (ReadLink(), Write
 * Link()).
 *
 * Returns 0 on success and a negative error code otherwise.
 */

int PacketMode (LinkId, PacketSize)
int LinkId;
int PacketSize;

/*
 * StreamMode
 *
 * Switch back the PacketLink to the default stream operation mode.
 * The packet mode operation on the PacketLink is terminated defi-
 * nitely. The output queue may contain pending packets. The total
 * number of discarded packets is reported as the return value.
 *
 * To know for a certainty that there are no pending packets,
 * 'InfoPacket()' should be called before calling 'StreamMode()'.
 *
 * The firmware parametrisation (mode and packet size) is controlled
 * by 'OpenLink()' and 'PacketMode()'. They will also detect if the
 * firmware hangs and will report. In such a case the additionally
 * supplied program 's4diag' may be called to force a power-on reset
 * at the BBK-S4.
 *
 * Returns the number of packets scheduled (by 'PutPackets()') but
 * not yet transferred via link. The return value 0 signals that no
 * packet was wasted. Otherwise a negative error code will be returned.
 */

int StreamMode ()

/*
 * InfoPacket
 */

```

```

* Used to get detailed information about the status of the
* PacketLink. This information is a 'snapshot' of the current
* situation. So the data are not guaranteed to be consistent.
* Nevertheless the most important aspects of these informations
* are covered. Every change of data during the snapshot is to
* the safe side, that means only more InPackets, more free
* OutPackets and less pending output packets actually than
* reported.
*
* The packetInfo structure contains the following components:
*
* struct packetInfo {
*     int LinkId; 'LinkId' of current PacketLink (-1 if none)
*     int PacketSize; Maximum packet size, see 'PacketMode()'
*     int MaxBuffSize; Maximum buffer size. Cannot write more
* bytes at once
*     int InPackets; Number of packets ready to 'GetPacket()'
*     int OutPackets; Number of packets free in 'PutPackets()'
*     int OutPending; Estimate on number of pending output packets
* }
*
* Returns a pointer to a (static) 'packetInfo' structure or NULL if
* there is no PacketLink.
*
*/

```

```

struct packetInfo *InfoPacket ()

```

```

/*
* GetPacket
*
* Read a single packet from the PacketLink into the buffer. The
* buffer pointed to by 'Buffer' must be large enough to hold a
* packet of maximum size (not checked). The start address of
* 'Buffer' has to be long alligned.
*
* This function is strictly non-blocking and will return very fast.
* There is no waiting of any form.
*
* Returns the number of bytes transferred or a negative error code.
* In case of no available packet ready to transfer, 0 is returned.
*/

```

```

int GetPacket (Buffer)
char *Buffer;

```

```

/*
* PutPackets
*
* Write as much packets to the PacketLink as possible. Each packet

```

```

* is described by an element of 'Packets', which is structured as an
* array of packets. 'Count' specifies the maximum number of valid
* packets to send.
*
* A packet is structured as follows:
*
* struct packet {
*   byte length; Packet header: bits 0..6: packet body length (bytes)
*   + 4 byte (pxhdr)
*       bit 7 : tag bit
*   int32 pxhdr; Parix header: 2 byte processor id, 2 byte channel no.
*   byte *body; Packet body
* }
*
* The start address of 'body' has to be long aligned. The best per-
* formance will be achieved by setting the body length to multiples
* of long words (int32).
*
* The operation is strictly non-blocking. There is no waiting of any
* kind. Each packet is tried to be transferred via the PacketLink as
* long as enough space is left on the BBK-S4 for a direct transfer.
* The number of directly transfered packets is returned. If it is not
* possible to send even a single packet, the return value is 0.
*
* Returns the number of packets transfered or a negative error code.
*
*/

int PutPackets(Packets, Count)
struct packet *Packets;
unsigned char Count;

/**** eof 'link.doc' ****/

```


Chapter 9

Troubleshooting

If you have any problems while using a link adapter please refer to this chapter first and try to find an appropriated section. If you can't find a description of your specific problem or you can't solve it please contact the service. In this case, please be prepared to give a detailed description of your problem. You may contact the service as deccribed in section 6.5.

9.1 No Light on Front LEDs (BBK-SCSI)

description: there is no light on front LEDs

possible reasons:

- The BBK-SCSI has no power supply.
Please check the connections of the power line, the position of your power switch an the backplane and the power supply of your outlet.
- The BBK-SCSI has not the right firmware inside it's Flash PROMS or the Flash PROMS are defect.

Try to install a new firmware using the command *burn_scsi* described in section 7.3.2. If the result is unsatisfactory or you don't have an up-to-date firmware version please contact the service for help.

9.2 Error Message: Failed to get existing sem_key

description: any tool using a link adapter brings the message: Failed to get existing sem_key.

possible reasons:

- the semaphore with the key: 0x42424242 does not have permissions for read and write for you. Please check first if the semaphore with this key is created by the group *vmebus*. If not, ask your administrator to erase this semaphore. If it is owned by the group *vmebus* please check if you are member in the group *vmebus*. If not, please ask your administrator to enter you into this group.
- the semaphore with the key: 0x42424242 is used by other programs. Ask your administrator how to avoid this problem.

9.3 Errors while Reading the Config File

description: There are problems while reading your config file. This file should be in the directory */usr/etc/transp* and is named *trans_conf*. It describes the hardware configuration of your host. You get error messages like:

Cannot read config file

Link adaptor name incorrect

Cannot find (stat) config file

Cannot get group-id for *vmebus/SBus* group

The config file is not member of the vmebus group
The config file must not have write permission for other
Cannot open config file for reading

possible reasons:

- Probably, the installation of your link adapter did not work correctly. Try to repeat the installation procedure or contact the service for support.

9.4 Errors within the Description of Your Hardware

description: There are problems while processing your config file. You get error messages like:

Syntax error in config file

Board number defined multiple in config file

The board-type name is not correct

This link adaptor does not exist

possible reasons:

- The installation of your link adapter did not work correctly or changes made in the config file have a wrong syntax. Check the entries of your trans_conf-file in the directory /usr/etc/transp and compare them with the examples given in the first section of this file.

9.5 Error Messages about Incorrect Parameters

description: You get error messages about incorrect parameters. They look like:

The board-number is out of range

The link-number on this board is out of range

possible reasons:

- You have started your program with parameters pointing to hardware (boards or links) which are not described in the file `trans.conf`. If the hardware exists, add the appropriated entries inside of this file accordingly to the examples on top of this file and section 3.2.3.3. If the hardware does not exist, use parameters which are in the valid range.

9.6 A Link is Showed as 'in use' but Nobody Uses it

description: The tools `ulti` and `links` show a link as 'in use' but you can't find someone using this link.

possible reasons:

- One application using this link has crashed. Sometimes it may happen, that it does not release the link. To clean up this situation, you must kill all processes which still use this link. If you use `PARIX` this is usually a `dserver`. Do a `ps -aef | grep dserver (SUNOS)`, `ps -aux | grep dserver (SOLARIS)`, respectively, and send the signal `TERM` to this processes by typing `kill pid` where `pid` is the process id gotten by the `ps`. If there are working more users please take

care that you kill only processes from your links! (You can figure out the link by the argument `-l link_number` given to the `dserver` and showed by `ps`.)

If you can't clean up the link by using this command, try to repeat it after turning the BBK-SCSI off and on or reset the BBK-S4 by the S-Bus tool `s4diag -r board-no`. If you still get a result send the signal `INT` to the processes by typing `kill -INT pid`.

Please notice that it may take a while until the process reacts on your signal if it is running inside of the system via the device driver!

9.7 Error Message: Access to link hardware failed

description: Any tool using a link adapter brings you the message: Access to link hardware failed.

possible reasons:

- Your link adapter is turned off. Please turn it on and try again.
- The system is not configured for the requested link or a SCSI-id is not entered correctly. Refer to section 3.2.3 to set up your system correctly.
- Your machine was booted without connected link adapters or with turned off BBK-SCSIs. You should boot your machine with turned on and connected link adapters.

Chapter 10

Tips and tricks

If you strictly follow the recommended installation instructions you will hardly get any trouble. The most frequent problem comes from semaphores with wrong group membership. The command

```
ipcs -sb
```

reports the id, the creator's user-id and group membership and the number of semaphores attached to this semaphore key. For each link-adaptor one semaphore is required. One additional semaphore is used during the initialisation phase of this semaphore-set. If the group membership is not `vmebus`, something went wrong. The first usage of the link-interface creates the semaphore-set. This is usually done during UNIX startup-time through `ulti -c`. The `ulti` command is installed by default inside `/usr/bin`. It is owned by 'bin' and member of the `vmebus` group. The `set-uid` and `set-gid` bits are set. Please check by typing

```
ls -lg /usr/bin/ulti
```

If the group membership seems to be ok, the number of attached se-

maphores may be wrong. This may be the result of changing the file */usr/etc/transp/trans_conf*. As mentioned before, this file is the database for the type and number of transputer boards active inside this machine. Installing an additional board requires the machine to be powered down. This will obviously clear any existing semaphores. Calling the link-interface for the first time will create the semaphore-set with the required number of semaphores. This number is computed from scanning the *trans_conf* file. If you change this file while a semaphore-set exists, the required number of semaphores may increase. The attempt to open the existing semaphore-set results in a rejection, due to missing semaphores.

Anyway it is possible to remove the existing semaphore-set, following these instructions:

First find out the id of the existing semaphores. The command

```
ipcs -sb
```

will report the id as the first item. Second this **sem-id** must be used as a parameter to the **ipcrm** command. The kernel chosen values start at 0 incrementing with 10. So let us assume the current sem-id is zero. Then

```
ipcrm -s 0
```

will remove the existing semaphore-set. Now apply all necessary changes to the *trans_conf* file. Call **ulti -v** again as an ordinary user and check the newly created semaphore-set using **ipcs -sb** again.

Be aware that the specification or the sbus device in *lconf.c* must not be changed. The slot number as specified in *trans_conf* is automatically appended by the BTSS software in order to enable multiple boards on a SBus.

Index

- l link_number, 93
- ../PARSYTEC, 25
- .descr, 59
- .tar, 59
- /bin/make, 56
- /dev, 58
- /etc, 8
- /etc/config, 25
- /parix, 67
- /usr, 7, 9, 11, 12, 14, 15, 28, 30
- /usr/etc/transp, 90, 91
- /usr/etc/transp/trans_conf, 79
- /usr/etc/transp/ulti, 15
- /usr/kvm/sys/sun4?/conf, 24
- ?, 24
- .descr file, 57
- .tar files, 53
- PARSYTEC Standard Installation, 53

- 0, 58
- 25, 8

- /usr/ccs/bin, 56
- pid, 92

- access to link hardware, 14

- additional workstation, 68, 69
- AT&T, 2

- basic installation, 54
- BBK-SCSI
 - firmware, 13
- BBK-S4, 1, 3, 11, 16, 28, 33-37, 74, 75, 93
- BBK-SCSI, 1, 3, 6, 13, 15-19, 21-24, 26-31, 71-74, 89, 93
- BBK-V1, 1
- BBK-V2, 1, 3, 28, 39-41, 43, 45, 46
- BBK-V4, 1, 3, 14, 15, 39-41, 43, 45
- bin, 7, 10-12, 15, 30
- boot, 26, 28
- BTSS
 - configuration, 12, 14
 - installation, 5
 - tools, 71
- BTSS, 1, 2, 5-7, 9-12, 14, 15, 17, 22, 28-30, 34, 35, 46-48, 71, 73, 74, 77-79
- burn.scsi, 30, 71, 74

- cd, 25, 57, 61, 67
- configuration, 12, 14
- configuration file, 14
- configuration syntax, 14
- cs, 56

- default location, 7
- delivery check, 58
- description file, 59
- df, 56
- disk space, 7

- Electronic Link Switch, 47
- error detection, 89
- etc, 7, 9, 14, 28
- XPLORER, 35

- fast installation guide, 54
- firmware
 - burn-in, 74
 - version, 73
- firmware, 13
- form-factors, 40

- GC/PowerPlus, 17, 22
- GENERIC, 24
- getversion, 29, 30, 73
- GNU make, 56, 62
- group, 8
- group, 8, 10, 60, 61, 64

- hardware
 - installation, 33, 39
 - requirements S-Bus, 33
 - requirements Vme-Bus, 40
 - S-Bus, 33

- Vme-Bus, 39
- host, 7

- IBM, 2
- INMOS, 2
- install, 53, 68
- install.mk, 66
- installation
 - PARSYTEC standard, 53
 - additional workstation, 68
 - available products, 58, 62
 - basic, 54
 - BTSS, 5
 - BTSS questions, 9
 - check, 15
 - common, 53
 - delivery check, 58
 - detailed, 55
 - fast, 54
 - first step, 9
 - group, 64
 - hardware, 33, 39
 - overview, 5
 - owner, 64
 - patch, 69
 - path, 64
 - permissions, 64
 - problems, 70
 - process, 61, 62
 - product interdependencies, 59
 - re-, 54, 67
 - requirements, 6, 56
 - S-Bus, 35
 - second step, 14
 - software, 5
 - standard, 53

- start, 61
- steps, 5, 9, 55
- summary, 55
- support, 70
- third step, 12
- user interface, 62
- Vme-Bus, 46
- installation, 5, 53
- installation guide, 55
- installation program, 62
- installation requirements, 59, 60
- installation stuff, 68
- installation tools, 58
- interprocess communication facilities, 7
- intrduction, 1
- ipcrm, 14
- ipcs, 7, 14
- Iserver, 18
- kill -INT pid, 93
- kill pid, 92
- lconf.c, 79
- lconf.o, 77, 78
- link
 - check, 71, 72
- link adapter
 - configuration, 12, 14
 - type, 1, 10
- link hardware
 - access, 14
- link interface, 79
- link.doc, 79
- link.h, 30
- link.o, 30, 77, 78
- link_c, 79
- links, 11, 12, 29, 30, 71
- login: root, 57
- ls -l, 56
- make, 25, 54-56, 62, 67
- make install, 54, 61
- makefile, 30
- mkdir, 57
- mmap(2), 78
- more, 8
- MOTOROLA, 69
- MTM-SUN, 1
- MULTICLUSTER-3, 35
- NFS, 68
- NIS database, 8
- OCCAM, 2
- online documentation, 59, 63
- owner, 10, 61, 64
- packetmode, 77
- parsylink.o, 78
- PARSYTEC, 24
- password:, 57
- patch installation, 69
- path, 10
- permissions, 61
- polling, 78
- POWEREXPLORER, 17
- POWERPC, 2, 18
- problems, 89
- product category, 65
- product description, 59
- product interdependencies, 59

- product specific hints, 64
- product specific Makefile, 53
- product specific preparations, 59
- ps, 92

- rehash, 15
- reinstallation, 54, 61, 67
- rlogin, 67, 69
- rmt, 58
- rst0, 58

- S-Bus device, 36
- s4diag, 11, 75, 93
- S-Bus, 1, 3, 5, 6, 8, 10, 11, 13, 33, 35, 36, 74, 78, 79, 93
- scsi_tools, 29, 30, 73, 74
- SCSI-Bus, 3, 5, 6, 8, 10, 15, 17, 22, 73, 79
- semaphore
 - key, 90
- semaphore, 7, 13, 90
- semaphore group, 14
- semaphore key, 14
- semaphores, 7
- set, 57
- shutdown, 26, 28
- software delivery, 58
- software problem report, 70
- SOLARIS, 1, 2, 7, 18, 24, 26, 29, 30, 36, 53, 56, 58, 68, 73, 74, 92
- SUNOS, 2, 7, 18, 24, 29, 30, 36, 53, 56, 58, 68, 73, 74, 92
- SPARC, 2
- SPARCSTATION, 7

- SPR, 69, 70
- standard installation, 53
- streamer tape, 7, 56
- streammode, 77
- su, 57, 61, 67
- SUN, 0-2, 9, 11-14, 17-19, 24, 28, 39-48, 56
- SUN-4, 2, 7
- SunOSx, 29, 30, 73, 74
- superuser, 57, 61
- superuser permission, 56, 61
- support, 70

- tape
 - read, 57
- tape device, 7, 56, 58
- tar, 54, 58
- tar file, 59
- tcsh, 56
- tools
 - S-Bus, 74
 - SCSI-Bus, 73
 - Vme-Bus, 75
- tools, 71
- TOOLSET, 3, 18, 79
- trans_conf, 6, 9, 12-16, 28-30, 73, 90-92
- transp, 7, 9, 14, 28
- troubleshooting, 89

- ulti, 11, 12, 15, 16, 30, 71, 72
- uname, 7
- UNIX, 2
- user, 60

- Vme-Bus

- 24-bit, 47
- 32-bit, 47
- address-assignment, 40
- configuration, 48
- hardware, 39
- jumper settings, 40
- link adapter, 39
- requirements, 47
- VME-Bus, 1, 3, 5, 6, 8, 10, 11, 13, 39-43, 45-48, 52, 75, 78, 79
- vmebus, 7, 8, 10, 13, 14, 35, 36, 47, 91
- VMTM
 - configuration, 48, 49
- VMTM, 1, 3, 14, 15, 39-41, 43, 45, 47-52, 75
- VMTM configuration, 47
- vmtmconf, 75
- which make, 56
- workstation
 - additional, 68
- workstation, 68
- workstation network, 61
- workstation operating system, 63
- yellow-page, 8
- ypcat, 8

