



digital
vision
group

Installing TRIP on UNIX

TRIPsystem
Product Documentation

End User License Agreement

All rights to this software, its documentation and logotypes of the TRIP product family and software (altogether “Software”) supplied by DVG Operations GmbH (DVG) are exclusively owned by DVG.

The transfer of this Software, solutions or parts thereof requires the prior written agreement of DVG. Furthermore, the customer has the right to use licensed Software and / or process solutions supplied by DVG to the extent specified in his contract with DVG.

The free-to-use non-commercial version doesn't require a prior written agreement with DVG but such customers, organizations and/or third parties agree by using the software and / or solution of DVG to be strongly obliged to keep all rights to this software, documentation and logotypes of the TRIP product family absolutely un infringed and protected.

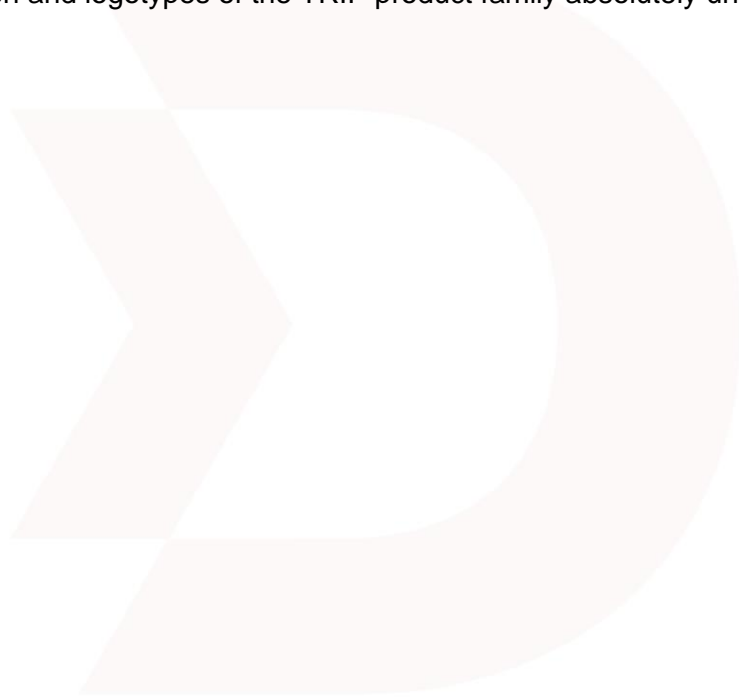


Table of Contents

PREFACE.....	5
1. PREPARING FOR INSTALLATION	6
1.1 PREREQUISITE HARDWARE AND SOFTWARE.....	6
1.2 PREREQUISITE DISK SPACE REQUIREMENTS	6
1.3 THE INSTALLATION PROCEDURE.....	6
2. INSTALLING TRIPSYSTEM	7
2.1 BEFORE RUNNING THE INSTALLATIONS SCRIPT	7
2.2.1 Loading TRIP	7
2.2.2 Preparing software dependencies.....	8
2.2.3 Remove TRIP services.....	8
2.3 STARTING THE INSTALLATION PROCESS	8
2.4 THE INSTALLATION MESSAGE.....	8
2.5 CHECKING FOR SOFTWARE DEPENDENCIES	9
2.6 EULA CONFIRMATION.....	9
2.7 THE TRIP DAEMONS.....	9
2.7.1 Stopping the daemons	9
2.7.2 Option to install the daemons for start via systemd (Linux only).....	9
2.7.3 Option to start tripd at end of installation.....	9
2.7.4 Option to start tbserver via xinetd (Linux only).....	10
2.7.5 Option to start tripnetd at end of installation.....	10
2.7.6 Configuration of tbserver on Solaris.....	10
2.8 SELECTING NEW OR UPGRADE INSTALLATION	10
2.9 TRIPSYSTEM FILE OWNERSHIP.....	11
2.10 SET SERVICE USERS	11
2.10.1 Choose tbserver user for xinetd and inetadm	11
2.10.2 Choose user and group for systemd services (Linux only)	12
2.11 DEFAULT LANGUAGE (NEW INSTALL ONLY).....	12
2.12 WARNING ABOUT OBSOLETE FILE PERMISSIONS.....	12
2.13 CONFIGURATION OF RUNTIME LINKER PATH	13
2.13.1 Configuration of ld.so (Linux only).....	13
2.13.2 Symbolic links from /usr/lib.....	13
2.14 CONTROL DATABASE SETUP AND MIGRATION.....	13
2.14.1 CONTROL database setup for new install.....	13
2.14.2 CONTROL database setup during update install.....	13
2.14.3 CONTROL database migration during update install.....	14
2.15 UPGRADING DEBIT.LOG, PRC AND PRN FILES.....	14
2.16 SETTING THE TRIPD PASSWORD.....	14
2.17 THE TRIPSYSTEM CONFIGURATION FILE TDBS.CONF	15
2.17.1 Configuration for a new installation	15
2.17.2 Migration of configuration during update install	15
2.18 PASSWORD FOR THE SYSTEM USER	15
2.19 NETWORK ACCESS CONFIGURATION	16
2.19.1 Listening port.....	16
2.19.2 Optional systemd service installation for tripd and tripnetd (Linux only).....	16
2.19.3 Optional xinetd service installation for tbserver (Linux only).....	16
2.19.4 Optional inetadm service installation for tbserver (Solaris only)	16
2.19.5 Optional start of the TRIP daemons without autostart	16
2.20 COMPLETION MESSAGES.....	16
2.21 ERROR MESSAGES	16
3. POST INSTALLATION REQUIREMENTS FOR TRIP	17
3.1 UPGRADING OLD DATABASE INDEX FILES	17
3.3 LOG FILES AND TEMPORARY FILES	17
3.3 VARIABLES FOR THE TRIP SERVER (TBSERVER).....	17

3.4 FIREWALL CONFIGURATION	17
4. THE TRIPDAEMON	18
4.1 AN OVERVIEW OF THE TRIPDAEMON	18
4.2 OPERATING THE TRIPDAEMON	18
4.2.1 <i>Summary of options</i>	19
4.2.2 <i>Starting the TRIPdaemon</i>	19
4.2.3 <i>Current information (-d file)</i>	19
4.2.4 <i>List current clients (-c)</i>	20
4.2.5 <i>Stopping the TRIPdaemon (-k)</i>	20
4.2.6 <i>TRIPdaemon logfile (-l)</i>	20
4.2.7 <i>Locked Record (-o)</i>	21
4.2.8 <i>Purge Clients (-P)</i>	21
4.2.9 <i>Current Queues (-q)</i>	21
4.2.10 <i>Restart daemon (-r)</i>	21
4.2.11 <i>TRIPdaemon Statistics (-s)</i>	21
4.2.12 <i>Log file Verbosity Level (-vn)</i>	22
5. REMOVING A TRIP INSTALLATION.....	23
5.1 ROLLING BACK A TRIP INSTALLATION	23
6. MANAGING TRIP IN A UNIX ENVIRONMENT	24
6.1 PATH DEFINITION	24
6.2 FILE PROTECTION	24
6.3 ENVIRONMENT NAME SPACES	25
APPENDIX A.....	27
SAMPLE INSTALLATIONS.....	27
<i>New Installation</i>	27
<i>Upgrade installation</i>	29
APPENDIX B.....	34
TRIPSERVER INIT FILE	34
APPENDIX C:.....	35
TROUBLESHOOTING.....	35

Preface

This guide contains information necessary for a UNIX system manager to effectively install a running TRIP system.

Please read this manual carefully before starting any of the processes detailed herein, as many of these processes can adversely affect the smooth running of a TRIP system if precautions are not taken.

Throughout this guide, there are references to a TRIP config file. This file is found in the installation directory (<TRIPpath>, defined in chapter 2.1) named as .TRIPrcs.template after a successful installation. When a system wide installation is done, this file is also copied to the <TRIPpath>/conf directory with the name tdbb.conf.



1. Preparing for installation

This chapter describes the steps that you should go through prior to attempting the installation of TRIPsystem. Please ensure that you have a complete distribution kit consisting of:

- This manual
- TRIP distribution media for the current version

1.1 Prerequisite hardware and software

To make a successful installation of TRIP possible, you will need to install TRIP on one of the supported UNIX platforms. Please refer to the latest Release Notes for TRIPsystem for an up-to-date list of supported platforms.

1.2 Prerequisite disk space requirements

To start the TRIP installation script, you need:

- 65 Mb free disk space on the installation device.

In addition, if you are performing a TRIP upgrade installation, you will need additional space on the installation device to accommodate the system CONTROL file and the system DEBIT file (if applicable) from your existing installation. Thus, for an upgrade installation, the space requirements are:

- 65 Mb + size of (CONTROL.*) +
size of (DEBIT.LOG) for the installation and target devices.

1.3 The installation procedure

You need to be a privileged user to run the installation procedure. The TRIP installation will ensure that you are privileged before allowing the installation to proceed.

The installation procedure performs a variety of different tasks. You will be prompted for a variety of responses during the installation to guide the process. For any of the queries, you may press "?" for more help or refer to this document for a more in-depth explanation.

The installation procedure will first ask a number of questions but the actions taken as a result of your answers will occur later, accompanied by messages telling what is done. The following paragraphs list the questions in the order they are asked. All of the installation messages are not listed here, but the most common ones may be found in the example installations listed in Appendix A.

2. Installing TRIPsystem

This chapter explains how to install TRIP and what prompts will be seen during the installation process. The order of the prompts in this section is not necessary that in which they appear at installation time. For up-to-date sample installation logs, refer to Appendix A.

2.1 Before running the installations script

2.2.1 Loading TRIP

Login as user “root” and change your present working directory (pwd) to the ‘top-directory’ where you want TRIP installed. Beneath that directory, the installation procedure will create a directory tree as shown below:

```
top-directory
|-- version-name
    |-- ase                # ASE library sample code
    |-- bin                # TRIPsystem programs
    |-- conf               # Configuration files
    |-- demo               # Demo databases
    |-- include            # TRIPtoolkit API header files
    |-- lib                # TRIPsystem shared libraries
    |-- prc                # Printer configuration files
    |-- sbin               # Privileged-access programs
    |-- share              # Miscellaneous files
    |   |-- conf           # Configuration template files
    |   |-- systemd        # SystemD service files
    |-- sys                # System resource files
    |-- trm                # Terminal definition files
```

Figure 1: The TRIP installation directory structure

The above schematic uses 'version-name' to refer to the directory named after the maintenance release number of the TRIP system being installed, for instance 'v821' for version v8.2-1.

Data and log files are from version 8.2 or TRIPsystem kept separate from the installed TRIPsystem files. The default directory for such data, as created by the installation procedure, is /var/trip on Solaris and /var/lib/trip on Linux:

```
/var[/lib]/trip
|-- ctl                # CONTROL database (TDBS_CTL)
|-- db                 # User databases (TDBS_BASES)
|-- log
    |-- sys            # TRIPsystem log files (TDBS_LOG)
```

Figure 2: The TRIP data files directory structure

Copy the tar file for the selected platform from the distribution media to your UNIX machine. Put the file in the directory (<top-directory> in Figure 1) where you want to build the TRIP installation. The name of the tar file may differ depending of the UNIX platform. Make sure you copy it in binary mode.

Now, run tar, e.g.

```
tar xvf tripsystem_rhel8-x86-64_v820.tar
```

This will create a directory structure as shown in Figure 1.

2.2.2 Preparing software dependencies

TRIPsystem has a number of dependencies on software that must be present on the machine before the installation proper can begin. While this is checked as part of the installation procedure, you can and should check this separately first.

To check for the presence of software dependencies, run the dependencies script as root:

```
<version-name>/sbin/dependencies
```

If the script reports one or more missing dependencies, you can on Linux choose to tell the script to install the missing dependencies unless you wish to do so separately:

```
<version-name>/sbin/dependencies --install
```

Note that using the dependencies script to install missing dependencies is only supported on Linux. For other UNIX operating systems, you must install the missing software packages yourself using the package manager of the operating system.

2.2.3 Remove TRIP services

If you have configured tripd, tripnetd and/or tbserver to launch automatically at boot using any other means that what is provided by the TRIPsystem installer, you must remove that custom service configuration before starting the trip installer. Keeping such service configuration intact may result in installation failure.

2.3 Starting the installation process

With current directory set to the <top-directory> (see Figure 1), the <version-name> directory or the sbin directory, you begin the installation procedure by running the install script:

```
<version-name>/sbin/install
```

where <version-name> is the TRIP version you want to install. For example:

```
v820/sbin/install
```

2.4 The installation message

The initial message that can appear once the installation procedure has begun relates to the kit that you are about to install. For instance, this message might tell you that this kit is hazardous to install while users are logged into the TRIP system. Alternatively, it may inform you that you have been provided with a pre-release kit, and to whom questions and reports should be directed. In most cases the installation message is empty.

Responding to all prompts should consist of entering a valid reply and then pressing <return>. Default responses are indicated in square [] brackets.

Following the message, the procedure prompts you:

```
Beginning installation of TRIP v8.2-0
Do you wish to continue? [Y|n?]
```

If you accept the default answer (by hitting RETURN), or type the affirmative response Y, the installation will continue. Otherwise, the installation will quit. During the installation process, a log file is created that will record all messages that occur. This file is located in the <TRIPpath> directory and is called *install.log*.

2.5 Checking for software dependencies

Like you are encouraged to do separately (see section 2.2.2), the install script will verify that all software dependencies are present. If all dependencies are present, the installation procedure will continue.

If one or more dependencies are found to be missing and the installation is being done on a RedHat Enterprise Linux, the install script will ask if it should attempt to install the missing software dependencies:

```
Do you wish to attempt to install the missing dependences? [Y|n]
```

If you decline by pressing N, the installation procedure will exit. If you enter the affirmative Y, the DNF package manager will be executed to install the missing packages. If this installation succeeds, the installation procedure will continue. If it fails, the installation procedure will exit.

2.6 EULA confirmation

If you have not installed this particular version of TRIP before on the current machine, the installation procedure will display an End User License Agreement (EULA) text and ask you if you agree to its conditions.

```
Do you accept these conditions? [Y|n|?]
```

Should you not accept, the installation procedure will exit.

2.7 The TRIP daemons

NOTE: If you have configured tripd, tripnetd and/or tbsever to launch automatically at boot by other means, you must remove that custom service configuration before starting the trip installer. Keeping such service configuration intact may result in installation failure.

2.7.1 Stopping the daemons

The installation requires that the TRIP daemons (tripd and tripnetd) are stopped. If they are running, they will be stopped automatically by the installation procedure. If the daemons previously have been configured to launch via SystemD (Linux only), the services will also be removed.

2.7.2 Option to install the daemons for start via systemd (Linux only)

If you are running a Linux system on which systemd is used, the installation script will ask if the TRIP daemons should be set up to start automatically via system:

```
Do you wish to start the TRIP daemons via systemd? [Y|n|?]
```

If you accept, the installation procedure will set up tripd and tripnetd for launch via system before it completes. Even if you decline at this stage, you can still configure tripd and tripnetd for launch via systemd at a later stage by running the script `install_services` that you can find in the `sbin` directory.

2.7.3 Option to start tripd at end of installation

If you are not running Linux or if you declined to configure the daemons for systemd in section 2.7.2, you will get a prompt asking if you want the TRIP daemon started automatically at the end of the installation:

The TRIP daemon (tripd) can be started at the end of the installation. Note that this doesn't install tripd for automatic start at boot. Do you wish to start tripd? [Y|n|?]

The default for this question is Y if the TRIP daemon was running when the installation script was started, and N if it was not.

Answering Y will make the installation procedure start the TRIP daemon at the end of installation. If you answer N the TRIP daemon will not be started and you will have to do this manually at a convenient time of your own choice. TRIP will not be available until the TRIP daemon has been started.

2.7.4 Option to start tbserver via xinetd (Linux only)

If you are installing TRIPsystem on a Linux machine where the xinetd superserver is available, and you have declined to start the TRIP daemons via systemd, the installation script will ask if the tbserver shall be launched via xinetd.

```
Do you wish to launch tbserver via xinetd? [Y|n|?]
```

If you accept, the installation procedure will configure networked access for TRIP to be managed via xinetd. In this case, there is no need to run tripnetd (and indeed, running tripnetd if tbserver has been configured via xinetd is likely to cause issues with accessing TRIP via the network).

2.7.5 Option to start tripnetd at end of installation

If you haven't opted to launch the TRIP daemons via systemd, nor having chosen tbserver start via xinetd, the installation script will ask if you want the TRIPnet daemon (tripnetd) started automatically at the end of the installation:

The TRIPnet daemon can be started at the end of the installation. Note that this doesn't install tripnetd for automatic start at boot. Do you wish to start tripnetd? [Y|n|?]

Answering Y will make the installation procedure start the TRIPnet daemon at the end of installation. If you answer N, the TRIPnet daemon will not be started and you will have to do this manually at a convenient time of your own choice.

TRIP network access will not be available until the TRIPnet daemon has been started, or the tbserver configured for launch via (x)inetd or systemd.

2.7.6 Configuration of tbserver on Solaris

When installing TRIPsystem on the Solaris operating, the installation script configures tbserver for automatic start using inetadm command unless you have chosen to start tripnetd at the end of the installation (see section 2.4.5).

NOTE: When upgrading TRIPsystem on Solaris, you should NOT choose to start tripnetd (see section 2.7.5) if your previous installation configured tbserver to launch via inetadm. Choosing to start tripnetd at this point in such a scenario is likely to cause unstable behavior when accessing TRIP via the network.

2.8 Selecting new or upgrade installation

The installation procedure will perform different functions depending on whether it is a new installation or an upgrade of an existing installation. In summary, an upgrade installation will do the following steps in addition to those performed for a new installation:

- Upgrade your CONTROL file to the new version

- Migrate your configuration file (tdbs.conf) to the new version
- Upgrade your printer control files to the new version
- Upgrade your TRIPclassic terminal definition files to the new version
- Preserve old accounting records, in DEBIT.LOG, if located under the TDBS_SYS directory.

If the installer has detected a TRIPsystem installation already present on the system, it will ask if an upgrade is to be performed or if the previous installation should be ignored and this should be a new install (without migrating previous data and configuration):

```
Is this a [N]ew installation or an [U]pgrade [U]?
```

Selecting to upgrade will direct the installation to perform those steps detailed above in addition to the normal installation steps. Selecting a new installation will require the process to perform a new installation only.

It is strongly recommended that you accept the default, as it is possible to partly destroy an existing installation by making the wrong type of installation. If you choose not to accept the default you will get an extra prompt asking if you are sure:

```
Are you sure? [Y]
```

Accepting the default [Y] will perform the chosen installation, any other answer will make the original prompt appear again and this will be continued until you have either accepted the original default or answered Y to the last prompt.

2.9 TRIPsystem file ownership

If you are performing an upgrade installation, the installation procedure will detect the owner and group assigned to the TRIPsystem files of the version being upgraded from.

If you are performing a new install, you will be prompted to set the owner of the TRIP files. The default is to set the owner to root.

```
Which user should own the TRIP files [root]?
```

If you enter an invalid name, you will be prompted again until a valid response is entered (either accepting the default or entering a valid user name).

In addition, you will be prompted to set the group owner of the TRIP files. There is no default. If you press return, you will be prompted again until a valid group is entered.

```
Which group should own the TRIP files?
```

If you enter an invalid group name, you will be notified that the name is invalid and prompted to re-enter a group.

2.10 Set service users

2.10.1 Choose tbserver user for xinetd and inetadm

When performing a new install and the installation procedure has determined that tbserver will be configured for launch via xinetd (Linux) or inetadm (Solaris), you will be prompted to set the user who will run the TRIPserver (tbserver). The default user is set to root.

```
Which user should run the TRIPserver (tbserver)? [root]
```

If you enter an invalid name, you will be prompted again until a valid response is entered (either accepting the default or entering a valid user name).

2.10.2 Choose user and group for systemd services (Linux only)

If you have chosen to start tripd and tripnetd automatically via systemd, the installation procedure will ask for the user and group as which to run tripd and tripnetd.

```
Which user should run the TRIP daemons? [root]
```

The default user will be the same as the chosen owner of the TRIPsystem files (section 2.6).

```
Which group should run the TRIP daemons? [root]
```

The default group is the same as the one chosen for the the TRIPsystem files (section 2.6).

NOTE: It is critically important to select this user and group so that operating system user and/or group will have write-access to all database files and other relevant directories and files for TRIP. Selecting a user+group combination with limited access to TRIPsystem resources will result in file privilege errors when running networked TRIP sessions.

2.11 Default language (new install only)

When running TRIP, a user can choose which language TRIP should use to communicate with them; likewise which language they will use to communicate with TRIP. The installation procedure prompts for which default language to use:

```
What is the default language when running TRIP?
[E]nglish/[S]wedish/[G]erman/[F]innish/[C]hinese/[N]orwegian: [E]
```

The response is applied to the TRIP config file variable TDBS_LANG so that during any given TRIP session where a specific language is not defined, this will be the language used by TRIP and by the user.

The currently supported languages are shown in the following table. Please supply the abbreviation in response to the prompt. Any other language choice or just pressing <return> will cause the language to default to English.

Language Name	Abbreviation
English	E
Swedish	S
German	G
Finnish	F
Chinese	C
Norwegian	N

The TRIP config file TDBS_CHARS (defining the choice of Character Set) will be set to LA1 except for Chinese, where it will be set to GBK.

2.12 Warning about obsoleted file permissions

Previous TRIPsystem versions normally applied the group s-bit for the TRIP executables. This option was used to ensure that TRIP would be able to manage the access rights to TRIP databases. This was important when TRIP as used as a server-side-only application without network access (using TRIPclassic only). This practise has been discontinued since it is no longer

relevant because TRIP is now normally used via the network and would require root privileges for tripnetd/tbserver. The best practice is now instead to set file permissions correctly (see section 2.7).

If the installation procedure discovers that you are upgrading from a TRIPsystem version in which the Set-Group-Id bit was set for the TRIP binaries, it will display a warning text. If you see this warning, read the text carefully then press the Enter to continue.

2.13 Configuration of runtime linker path

2.13.1 Configuration of ld.so (Linux only)

When installing TRIPsystem on Linux, the installation procedure will attempt to edit the ld.so linker configuration with the directory in which the TRIP kernel library libtdbs.so is located. This step will typically create a file tripssystem-libpath.conf in the /etc/ld.so.conf.d directory.

If successful, the operating system is now using the /usr/local/trip/sys/lib directory as an additional standard directory for shared libraries. This allows custom server-side programs and ASE libraries that utilize the TRIPtoolkit C API to locate their libtdbs.so dependency correctly.

When linking TRIPtoolkit programs and ASE libraries, it is on Linux not sufficient to specify the libtdbs.so link dependency by stating "-ltdbs". This is because of differences in how ld.so behaves and the software linker ld behaves when building executables. Link directives for TRIPtoolkit programs must therefore always be specified as follows on Linux:

```
-L/usr/local/trip/sys/lib -ltdbs
```

Just using -ltdbs will result in linking error.

2.13.2 Symbolic links from /usr/lib

When installing TRIPsystem on Solaris as well as in the event that the configuration of ld.so failed when installing on Linux, the installation procedure will create a symbolic link to the libtdbs.so library from the /usr/lib (and /usr/lib64, where relevant).

2.14 CONTROL database setup and migration

The CONTROL database and its template files (P_CONTROL) are expected to be located in the directory referred to in the tdb.conf configuration file as TDBS_CTL. The standard value for this is from TRIPsystem 8.2 and later /var/lib/trip/ctl on Linux and /var/trip/ctl on Solaris.

2.14.1 CONTROL database setup for new install

When performing a new install, the control database files CONTROL.BAF/BIF/VIF and the control database template files P_CONTROL.BAF/BIF/VIF are copied to the standard TDBS_CTL directory.

2.14.2 CONTROL database setup during update install

When performing an update installation such that the old TDBS_CTL directory is different from the new one, the installation procedure will display the following message (example):

```
Copying <OLD_TDBS_CTL>/CONTROL.* to
      <NEW_TDBS_CTL>/CONTROL.*
```

If the TDBS_CTL directory in the new version is the same as in the old, the installation procedure backs the existing CONTROL database files before proceeding.

2.14.3 CONTROL database migration during update install

The migration of the CONTROL database is a mandatory and critical step during an update installation of TRIPsystem. During this migration, the CONTROL database will be brought up to date with the TRIPsystem version being installed by using the template control database files (P_CONTROL) as master. Any existing users, databases, procedures, etc, will be retained.

The control database migration is handled by a subordinate script named "migrate_control", which performs the following steps:

1. If there are ten or more pre-existing backups of the CONTROL database, the oldest one will be removed.
2. The PACKIT program is run on the CONTROL database. This makes sure that its structure is as efficient as possible.
3. The CONTROL database is fully reindexed.
4. The main CONTROL database migration tool, MODCON, is executed.

All the four steps show a varying degree of output. If any errors occur during the CONTROL database migration, this error output will be useful help to diagnose the problem. If an error occurs, the installation procedure will halt prematurely.

When the CONTROL database migration has been successfully completed, the installation procedure prints the following message:

```
The CONTROL database migration completed successfully.
```

2.15 Upgrading DEBIT.LOG, PRC and PRN files

As part of the update install, the installation procedure will copy any pre-existing DEBIT, PRC and PRN files used with the previous installation for continued use with the new installation.

If a DEBIT.LOG file exists in the current <TRIPpath>/sys directory, then it will be copied to the new <TRIPpath>/sys directory. If, however, a DEBIT.LOG file exists elsewhere, then the file will NOT be copied to the sys directory of the new version. A message will appear:

```
A DEBIT.LOG file exists in <directory>. This will NOT be copied
automatically.
```

where <directory> is the directory pointed to by TDBS_ACCDIR in the old TRIP config file and it will be the system manager's responsibility to set up the accounting for the new version.

If any printer control files (.PRC and .PRN) are used with the current version of TRIP, these files will be copied to the new prc directory provided TDBS_PRC is defined in the TRIP config file. This will produce the message.

```
Copying PRC and PRN files from <current-version> to <new-version>
```

2.16 Setting the tripd password

From version 8.0-0, using a password to start and control the tripd is no longer mandatory. This part of the installation procedure has therefore been removed from the installation program. Should the installation procedure detect that a password has been registered with the TRIP version being upgraded from, the following message will be displayed:

```
***** Found an old TRIP daemon password file in <OLD_TDBS_EXE>
This behavior is obsoleted and no longer recommended.
The password file will therefore not be migrated.
```


2.17 The TRIPsystem configuration file tdbb.conf

The location of the TRIPsystem configuration file tdbb.conf is the <TRIPpath>/conf directory. Users of TRIP on Docker will notice that this differs from Docker installations of TRIPsystem where it for practical reasons is located in /var/lib/trip/conf/sys.

2.17.1 Configuration for a new installation

When performing a new installation, the tdbb.conf file will be initialized with default values. Depending on how you intend to use TRIP, you may wish to add additional configuration parameters. Refer to the TRIPsystem Environment document for details.

2.17.2 Migration of configuration during update install

When performing an update installation, the previous tdbb.conf file will be copied into the conf directory of the new installation and adjusted to fit the new TRIPsystem version. The following adjustments are made.

Symbol	Adjustment
TDBS_LOG TDBS_APILOG TDBS_SCRATCH TDBS_SIF	If the values of these symbols refer to the log directory under the old installation, they are changed to the /var/lib/trip/log/sys directory on Linux and to /var/trip/log/sys on Solaris.
TDBS_PRC	Changed to <TRIPpath>/prc
TDBS_TRM	Changed to <TRIPpath>/trm
TRIP_DEMO	Changed to <TRIPpath>/demo
TDBS_CONFLATORS	Changed to <TRIPpath>/lib/libtripstem.so
TDBS_XMLSHR	Removed since the TRIPxml functionality is now built into TRIPsystem.
TRIPSQL_ENGINE	Removed since the TRIPsql functionality is now built into TRIPsystem.
TDBS_BASES	If not previously defined, it is set to the /var/lib/trip/db directory on Linux and to /var/trip/db on Solaris. The directory will be created if it does not already exist. Full access will be granted to the specified owner and group of the TRIP installation, and read access to all other users.
TDBS_EXE	Changed to <TRIPpath>/bin
TDBS_SYS	Changed to <TRIPpath>/sys
TDBS_CTL	If the value refers to the sys directory under the old installation, it is changed to the /var/lib/trip/ctl directory on Linux and to /var/trip/ctl on Solaris.

2.18 Password for the SYSTEM user

If you are performing an update installation, the SYSTEM user's password will be left intact. If you are performing a new installation, you will be prompted to provide and verify the TRIP SYSTEM user's password:

```
**** TRIP System Utility SETSPWD - Set TRIP System Password ****
Version 8.1-0   16-Mar-2021 08:49:12
```

```
New Password          :  
Verification          :
```

NOTE: Neither the password nor the verification will be echoed to your screen. If the verification does not match the supplied password, these prompts will loop.

2.19 Network access configuration

2.19.1 Listening port

The TCP port that the TRIP server listens to for network connections is configured as the “pctdb” service in the `/etc/services` file. The port number is 23457 by default, but if this port is already in use by another service and the `pctdb` service hasn’t already been defined, the installation procedure will prompt for an alternative TCP port number to use. It will suggest 23567 as an alternative, but you may choose any available number.

2.19.2 Optional systemd service installation for tripd and tripnetd (Linux only)

If you have opted to install `tripd` and `tripnetd` for launch via `systemd` when installing on Linux, this installation will now be performed.

2.19.3 Optional xinetd service installation for tbserver (Linux only)

If you are installing on Linux and have opted to configure `xinetd` to listen for TRIP network connections, servicing `tbserver`, this installation will now be performed.

2.19.4 Optional inetadm service installation for tbserver (Solaris only)

If you are installing on Solaris and have opted to configure `inetd` (`inetadm`) to listen for TRIP network connections, servicing `tbserver`, this installation will now be performed.

2.19.5 Optional start of the TRIP daemons without autostart

If you have opted to have the installation procedure start `tripd` and/or `tripnetd` at the end of the installation, this will now be done.

2.20 Completion messages

The installation of TRIP is now complete. You should now proceed to Chapter 3, “Post Installation Requirements for TRIP”, unless some error has occurred, in which case please refer to the next paragraph.

2.21 Error messages

If an error occurs during the installation, the error message will be output either in the displayed text from the installation procedure or in the installation log file, *install.log*, which is located in the `<TRIPpath>` directory (see section 2.1). You should, as the actual message indicates, attempt to correct the reported problem(s) before restarting the installation.

You can also use the script ‘`chkinst`’, that can be found in the ‘`sbin`’ directory of the installation, to check if your installation is correctly setup.

3. Post installation requirements for TRIP

3.1 Upgrading old database index files

If you have upgraded a TRIP installation from an older major version, it may be beneficial to re-index your databases in order to make sure that you can use the full range of all TRIP search features.

The Change History document will mention if there are any new features or corrected defects for which a re-index is recommended.

3.3 Log files and temporary files

The configuration variables **TDBS_LOG**, **TDBS_SCRATCH** and **TDBS_SIF** are all set up by the installation procedure to `/var/lib/trip/log/sys` on Linux and `/var/trip/log/sys` on Solaris when performing a new installation.

If you wish to use different locations for these variables, you may change the `tdbs.conf` file accordingly. Please remember to also set the directory permissions so that all operating system users who run TRIP processes (`tripd`, `tripnetd`, `tbserver` and `trip`) have write access.

For information about what these and other configuration variables are used for, refer to the TRIPsystem Environment document.

3.3 Variables for the TRIP server (tbserver)

If the **TDBS_SCRATCH** configuration variable is not defined, and the TRIP server process `tbserver` is started as root, the working directory will normally be set to `/` (root's login directory). To avoid this, the default working directory has been preset to `/tmp`.

The setting of this variable is critical if the disk partition where the directory `/tmp` is located is too small. If the temporary files cannot be created or enlarged, the `tbserver` is likely to fail.

NOTE: All temporary files created for index jobs activated by a client/server application using any of the products in the TRIPsdk family, will be placed under **TDBS_SCRATCH** or, if specified, under **TDBS_STO_LOCATION** and **TDBS_BUT_LOCATION**.

Please refer to Appendix C for checking up that TRIPsystem has been correctly installed and that network access is enabled for TRIP.

3.4 Firewall configuration

TRIPsystem includes a network service that listens on a TCP port defined for the "pctdbs" service in the `/etc/services` file. This is typically port 23457 or 23567, but may be customized if so desired. This port must be opened up in any firewall between the TRIPsystem server and machines on which the clients are running. Possible client software includes TRIPmanager, TRIPnpx and TRIPjxp as well as applications based on these, such as the TRIP Web Client.

4. The TRIPdaemon

4.1 An overview of the TRIPdaemon

The TRIP system includes a lock manager, called the TRIPdaemon. This manager is a memory resident process to which all TRIP users must connect during login. If such a connection cannot be made, the TRIP session will not start, and the user will be advised to contact the TRIP system manager to remedy this situation.

The TRIPdaemon process performs three functions:

1. To maintain a table of currently active TRIP users. This table includes the TRIP username, their UNIX username, and a unique session token. Based on the license you hold for TRIP, your site may be allowed only a certain number of tokens to be in use at any one time. In this case, once these tokens have been allocated to users, any further users who attempt to login will be given a message telling them that the current licensed limit of concurrent users has been exceeded and that they should try to login again later.
2. To maintain a table of currently locked records. This table includes, for each database that any user has open for write access, a list of locked records and pointers for each record into the TRIP user table. This means that when a user attempts to edit a record, the system will check with the TRIPdaemon to ensure that no other user is editing this record. If the record in question is locked, i.e. another user is currently editing it; a message is issued by the TRIPkernel informing the user of the status of the record, including the current lock holder's UNIX and TRIP usernames.
3. To start requested "batch" jobs like loading, indexing, globally modifying and printing database records.

When a user exits TRIP, by whatever means, the session token and associated locks held by that user will be automatically run down, thus ensuring that "stale" locks and sessions are not maintained by the TRIPdaemon.

4.2 Operating the TRIPdaemon

The TRIPdaemon (tripd) must be running in order for TRIP to be used. If it is not started, logins to the TRIP system will not be allowed, and every time users attempt to login, they will see the message:

```
TRIPdaemon failed to respond (0), notify your System Mgr.
```

In addition, if the TRIPdaemon is ever stopped, any users who are currently logged into TRIP will see the same message when they attempt to perform any operations within TRIP.

4.2.1 Summary of options

The following is a list of command line parameters available with the daemon. Detailed information can be found in the following sections.

-a	alter daemon password
-dfile	dump current information held to <i>file</i>
-c	list current users
-kn	kill the daemon in <i>n</i> minutes
-lfile	set the logfile to <i>file</i>
-o	list information about all current locked records
-P	purge clients
-q	list information about currently defined queues
-r	restart
-s	list daemon statistics
-vn	change verbosity level of logfile to level <i>n</i>

4.2.2 Starting the TRIPdaemon

Unless the TRIPdaemon (tripd) is configured to be started by systemd (Linux only), it must be started manually. To do so, enter the following:

```
<TRIPpath>/bin/tripd
```

The only option that may be specified when starting the daemon is *-l file* (refer to 4.2.6).

If you would like the daemon started up on boot up of the system and do not have access to systemd or do not wish to use it, place the call to tripd in a boot file.

Review specific start-up files and syntax for your system for other options.

NOTE: You must start the TRIP daemon as a user with full access to all TRIP database files and directories. If the user does not have sufficient access, TRIP will display unpredictable behavior.

4.2.3 Current information (-d file)

The -d option will dump the current information held by the daemon to a file that has been specified. For example, if the command:

```
<TRIPpath>/bin/tripd -d tripdump.txt
```

then the file tripdump.txt might contain information such as:

```
*****
QUEUES 1
QUE PRINT
LIM 0
IDL 0
UID 0
GID 0
STIME 748023258
NO_JOBS 0
*****
CLIENTS 1
PID 20610
UID 211
GID 201
TIME 748032547
NO_JOBS 0
NO_LOCKS 0
*****
NO_DBID 0
*****
```

4.2.4 List current clients (-c)

If you would like a list of currently logged clients, then you would use the -c option. For example, issuing the command:

```
<TRIPpath>/bin/tripd -c
```

would evoke a similar display such as:

PID	UID	GID	TIME	LOCKS	JOBS
15511	0	0	14:54:57	0	0
17299	218	202	14:51:55	1	0

Where PID is the process ID of a client, UID is the UNIX user ID and GID is the UNIX group ID of the person who owns the process. Other information includes the time of the process and whether they have locked any records and whether they have any jobs pending.

4.2.5 Stopping the TRIPdaemon (-k)

To stop the TRIPdaemon, i.e. shutdown the TRIP system, you should execute command:

```
<TRIPpath>/bin/tripd -k
```

Next you will be prompted for a delay time before shutdown:

```
* How many minutes to final shutdown:
```

You may also enter the shutdown time on the system line. For example:

```
<TRIPpath>/bin/tripd -k 2
```

4.2.6 TRIPdaemon logfile (-l)

As mentioned earlier, the only option that is available when starting the daemon is the -l option. This option allows for the creation of a log file of daemon events. The type and amount of information stored in the log file is controlled by the -vn option (section 4.2.13). To set the logfile, type:

```
<TRIPpath>/bin/tripd -llogfile.txt
```

If the log file was set up with the daemon, then it might contain information such as:

```
[05/02/22 18:44:16] TRIPdaemon v8.2-0
[05/02/22 18:44:16] -----
[05/02/22 18:48:59] dump_information: Dumped information to
'/tmp/EYaABFmAAA'.
[05/02/22 18:48:59] TRIPdaemon v8.2-0
[05/02/22 18:48:59] -----
[05/02/22 20:54:11] TRIPdaemon v8.2-0
[05/02/22 20:54:11] -----
[05/03/22 16:13:54] TRIPdaemon v8.2-0
[05/03/22 16:13:54] -----
[05/03/22 16:13:54] dump_information: Dumped information to
'/tmp/EmdABFmAAA'.
[05/03/22 16:14:07] TRIPdaemon v8.2-0
[05/03/22 16:14:30] -----
[05/03/22 18:50:58] dump_information: Dumped information to
'/tmp/EYNMdasATM'.
```

4.2.7 Locked Record (-o)

To list information about all currently locked records, you would use the -o option:

```
<TRIPbin>/tripd -o
```

For example, the result of the above command might show:

DBID	5				
OBJID		PID	UID	USEID	RECNO
2		30299	218	2	1

DBID is the database ID, PID is the process ID associated with the locked record, UID is the user ID of who has locked the record, USEID is TRIP user ID who has locked the record, and RECNO is the record number that is locked.

4.2.8 Purge Clients (-P)

This command will allow you to clean up clients that have been left hanging by incorrectly exiting TRIP. With the command:

```
<TRIPpath>/bin/tripd -P
```

4.2.9 Current Queues (-q)

To list information regarding any currently defined queues, you would use the -q option. The command line would be:

```
<TRIPpath>/bin/tripd -q
```

which would result in an output such as:

ST	QUEUE	CTIME	LTIME	RJOBS	PJOBS
R	PRINT	12:14:18	15:11:07	0	0

The ST column is the status of the queue, either R (running) or P (pending). The next column is the queue name followed by the connect time, the local time, the number of running jobs and the number of pending jobs.

4.2.10 Restart daemon (-r)

When you restart the daemon, it will tell the currently running daemon to die after it has dumped all information about pending jobs, queues, clients and locks to a file. There should be no batch jobs running or pending before restarting the daemon. The only reasons for restarting the daemon is to reset the daemon statistics, to change the verbosity level in the log file, or if upgrading to a new version. If you need to restart the daemon, it must be the last command in a line of arguments. For example, if you issue the command:

```
<TRIPpath>/bin/tripd -ltripd.log -v2 -r
```

You are telling the daemon to restart and when it does the verbosity level in the log file tripd.log will be 2.

4.2.11 TRIPdaemon Statistics (-s)

For general statistics regarding the daemon and its activities, such as queues, jobs, clients and objects, you can use the -s option. Issuing the command:

```
<TRIPpath>/bin/tripd -s
```

would result in a display as such:

```
TRIPdaemon v8.2-0, started 05/03/22 12:14:30

JOBS      QUEUED    FORKED    EXITED    KILLED    FORKERR
          20         20        20         0         0

QUEUES    CREATED    EXPIRED    KILLED
          7         7         0

OBJECTS   LOCKED     (ERRS)    UNLKED    (ERRS)    REMOVED
          22         0         19         0         3

CLIENTS   LOGIN      LOGOUT    REMOVED    DENIED
          34        33         3         0
```

4.2.12 Log file Verbosity Level (-vn)

The verbosity level of the log file can be set with the -v option. For this option, there are eight (8) levels available. These levels area:

- 1 Log vital error messages about privilege violations and internal error messages only
- 2 Log more error messages including objects not being locked and license limitation messages
- 4 Log vital information messages
- 8 Log all relevant actions performed by the daemon. This results in a lot of information
- 16 Reserved
- 32 Reserved
- 64 Log information about jobs being queued and jobs being started

The command for the verbosity level is:

```
<TRIPpath>/bin/tripd -v2
```

which would result in the log file containing information on objects not being locked and license limitations. Any of these levels can be combined. For example, if you would like information on jobs (64) and privilege violations (1), you would enter:

```
<TRIPpath>/bin/tripd -v65
```

You can also change the log file of a running daemon.

5. Removing a TRIP installation

The <TRIPpath>/sbin directory contains a script named “uninstall”. Contrary to its name, it doesn’t fully uninstall TRIPsystem. Rather, it deactivates the installation by:

- Removing the symbolic links /usr/local/trip/sys/lib and /usr/local/trip/sys/conf
- Optionally removing the contents of the TDBS_LOG directory
- Uninstalling the tripd and tripnetd daemons from systemd launch (Linux only)
- Stopping the tripd and tripnetd processes if they are running
- Removing the <TRIPpath>/lib directory from the dynamic linker

NOTE: After this script has been executed, you can no longer perform an update installation since your previous installation can no longer be detected. While your data will not be removed, any subsequent reinstallation of TRIP will require a manual migration of your data.

For a more complete removal of a TRIP installation, here is a list of steps to consider if a complete de-installation of TRIP should be needed:

- Stop the TRIP daemon process tripd
- Remove the link /usr/lib/libtdbs.* (also 64-bit links, if any)
- Remove the directory /usr/local/lib/trip (and ev. /usr/local/trip)
- Remove all TRIP settings from (x)inetd related files
- Then restart the (x)inetd process to run without references to TRIP
- Remove the TRIP config file and all copies of it
- Remove the TRIP installation directory recursively
- Remove any TRIP database files, file extensions .BAF, .BIF, .VIF
- Remove TRIP data files (.TFO, .def) and temporary files (e.g. .BUT, .STO, .TPO)
- Remove all TRIP settings from system boot or start-up scripts
- Remove all TRIP settings from user .profile, .login and start-up scripts
- Remove any log files kept in directories not yet removed (e.g. /tmp)
- Remove users/groups used only for running TRIP
- Remove TRIP servers from the /etc/hosts file (if not having other uses)
- Remove TRIP from the PATH variable and from other environment variables
- Remove environment variables only used by TRIP
- Remove disk mounts used only for TRIP purposes

5.1 Rolling back a TRIP installation

There is no automated way of rolling back a TRIP installation to an earlier TRIP version; it is a complicated procedure and it is very easy to destroy the TRIP installation, making it impossible to run TRIP. If you really need to roll back to an earlier version, please contact TRIP support for help.

6. Managing TRIP in a UNIX Environment

This chapter describes the three major issues concerning security and general management of a TRIP installation: path definition, file protections schemes and environment name spaces.

6.1 Path definition

To make it possible to start TRIPclassic by just entering the name of the executable, trip, the path to the bin directory of the TRIP installation must be entered into the PATH environment variable. This will also make it possible to start TRIP utilities, e.g. SETLOCK, in the same way.

This is best done by entering the following 3 lines into a common or user specific start-up file as e.g. /etc/profile, .profile, .login or .cshrc.

```
TDBS_EXE=`sed -n 's/^[ \t ]*TDBS_EXE[ \t ]*=[ \t ]*\([^#]*\) .*/\1/p'
/usr/local/trip/sys/conf/tdbs.conf | tail -1`
```

```
PATH=$TDBS_EXE:$PATH
```

```
export PATH
```

The installation process creates a TRIP config template file in the installation directory:

TRIPrcs.template a template name space definition file

NB! Please note that changing the PATH variable to include the path to the current TRIP installation requires that the PATH variable **MUST** be changed whenever the TRIP installation is upgraded or rolled back, otherwise you will try to run the wrong version of TRIPclassic..

The easiest way of making sure that the PATH variable contains the correct version will be to always logout and then login again after upgrading the TRIP installation, providing you have used the correct way of entering the path to the trip executable.

Entering the path to the trip executable in the PATH variable should be done as described above, i.e. getting the path from the TDBS_EXE variable in the TRIP config file. **DO NOT** enter a path explicitly into the PATH variable; it is very easy to forget to change it manually.

Also, **DO NOT** set up other TDBS or TRIP environment variables with the env, export, set or setenv commands, they should only be set in the TRIP config file, to avoid situations where you get a “mixed” installation, with some variables pointing at the current TRIP version and some pointing at the previous version.

It is of course possible to do the ‘DO NOT’s above, but then the important thing is that after a TRIP installation these variables must be checked to insure that they don’t point at the previous installation, they must point at the current one. The ‘DO NOT’s should be viewed as a kind of ‘best practice’ advice.

6.2 File protection

TRIP executables do not need to run as ‘root’ in order to be able to correctly manage the security of a database. To make this work correctly, the installation process prompts for the name of a group which will not only own the TRIP executables, but also must own *all* TRIP files (i.e. BAF/BIF/VIF/LOG).

Thus, suppose we decide to grant group ownership of TRIP to ‘tripgrp’, then the TRIP executables will appear as:

```
-rwxr-xr-x      root      tripgrp      .. baffit
```



```
-rwxr-xr-x    root    tripgrp    ..  lxit
-rwxr-xr-x    root    tripgrp    ..  trip
```

All TRIP database files must now be defined as owned by 'tripgrp' (for example). This is a manual process and must be performed by the manager before allowing users access to the new version.

```
-rw-rw-r--    root    tripgrp    ..  MYDB.BAF
-rw-rw-r--    root    tripgrp    ..  MYDB.BIF
-rw-rw-r--    root    tripgrp    ..  MYDB.VIF
-rw-rw-r--    root    tripgrp    ..  MYDB.LOG
```

Note that not only must the group ownership be set, but also the group access must be set to mode 6.

All operating system users who run TRIP processes must belong to the group (tripgrp in this example). For this reason, there is no requirement to grant any 'other' user access. Nor is there any requirement to have either the TRIP files or the executables owned by 'root'.

6.3 Environment name spaces

The environment name space of TRIP users is controlled by the TRIP config file: tdbb.conf which can reside in two places, the <TRIPpath>/conf directory and the user's home directory (optional). In the <TRIPpath>/conf directory, the file can be used to define names which are 'privileged' as well as 'unprivileged', whereas the file in the user's home directory can only be used to define unprivileged names (and thus override unprivileged definitions in the root file, but never to override privileged definitions). Note that environment definitions performed in the user's .profile or .cshrc file (or other startup files) count as non-privileged definitions and override any such definitions in the TRIP config files.

The format of the TRIP config file follows the familiar Windows .INI format :

```
[Privileged]
TDBS_ACCDIR=/var/lib/trip/log/acc
TDBS_ACCFLG=8

[NonPrivileged]
TDBS_CTL=/var/lib/trip/ctl
TDBS_CHARS=MUL
```

using named sections for the privileged and non-privileged definitions of names. The purpose of using the privileged section is to restrict access to those names which define privileged operations within the kernel (namely the accounting functions).

There can be three types of named sections within such a file:

```
[Privileged]
[Group:<group1>,<group2>,...,<groupn>]
[NonPrivileged]
```

Most names are read by the kernel at the outermost level, i.e. if defined in the non-privileged section, they will be read from there first, then from the group level and finally from the privileged level. Certain names can only be read from the Privileged section, however :

TDBS_ACCDIR	Directory for storing accounting files
TDBS_ACCFLG	Flags dictating the level of accounting performed
TDBS_SYS	Location of DEBIT.SYS (if TDBS_ACCDIR is not used)
TDBS_EXE	Location of TRIPD.DAT
RevokeWorldRead	Used to stop 'r' bit for world access to TRIP files when created

Note that the privileged definitions of TDBS_SYS and TDBS_EXE are only used when operating on privileged functions. For all non-privileged uses of these names, the outermost definition will be used, i.e. these names are normally defined at both privileged and non-privileged levels.

The last name is used by the file protection scheme, described in the previous section, to determine the protection given to new files created by TRIP :

```
RevokeWorldRead=FALSE    -rw-rw-r--
RevokeWorldRead=TRUE     -rw-rw----
```

The Group section (which can be repeated as many times as desired) is to allow names to be differently defined for different user groups. For example :

```
[Group:users,managers]
TDBS_CTL=/var/lib/trip/ctl
[Group:developers]
TDBS_CTL=/home/trip/v820/dev_ctl
```

so that if the calling process had membership of the group 'users' or the group 'managers', the name TDBS_CTL would be defined differently than if it had membership of 'developers'. If the calling process had membership of both 'users' and 'developers', however, the 'developers' definition would override as it is stated last in the file.

Names within the file can also use variable substitution to read from the user's environment. For example:

```
TDBS_SIF=/home/trip/sif/$LOGNAME
```

would define the location of SIF files as dependent on the user's login name. If the name being used for substitution contains characters other than alphanumeric and the underscore (_) or if you want to concatenate variables within the substitution, you can use curly braces to surround the name being substituted, for example:

```
TDBS_SIF=/home/trip/sif/${VAR(3)}$MYVAR
```

which would define TDBS_SIF as being dependent on the variable 'VAR(3)' concatenated with 'MYVAR'.

Note that entries within the TRIP config file are case sensitive.

Appendix A

Sample installations

New Installation

The following is an example of a new installation of version 8.2-0 of TRIPsystem on Enterprise RedHat Linux 8. This is the text displayed on the console/terminal:

```
#
# v820/sbin/install

PLEASE NOTE!
  When prompted for a response, entering a ? gives a short help text

Beginning installation of TRIP v8.2-0 on a Linux x86_64 platform.
Do you wish to continue? [Y|n|?] Y
```

```
***** Answer: Y
```

Continuing with the installation process

```
NB! Installation log file is:
    /opt/trip/sys/v820/install.log
```

Checking for presence of prerequisite software ...

```
bash: 4.4.20
sed: 4.5
libstdc++: 8.5.0
gawk: 4.2.1
binutils: 2.30
procps-ng: 3.3.15
psmisc: 23.1
file: 5.33
libgcc: 8.5.0
dmidecode: 3.2
libc: 60.3
openssl: 1.1.1k
expat: 2.2.5
zlib: 1.2.11
```

Good news! All prerequisites are already installed.

umask is changed to 022

```
***** Install TRIP v8.2-0 on a Linux x86_64 platform
```

```
2022-05-02 12:20:11
```

The TRIP daemons can be installed for automatic start via systemd.
Do you wish to start the TRIP daemons via systemd? [y|N|?] y

```
***** Answer: y
```

Proceeding with a new installation of TRIP

which user should own the TRIP files? [root]

```

***** Answer: default [= root]

which group should own the TRIP files? tripgrp
***** Answer: tripgrp

which user should run the TRIP daemons? [root]
***** Answer: default [= root]

which group should run the TRIP daemons? [tripgrp]
***** Answer: default [= tripgrp]

what is the default language when running TRIP?
[E]nglish / [S]wedish / [G]erman / [F]innish / [C]hinese / [N]orwegian
[E]

***** Answer: default [= E]

The TRIP language selected was ENG
( Setting the character set to LA1 )

Setting TRIP file protections
Setting TRIP file ownership to:
    "root:tripgrp"
Giving user "root" ownership of the TRIP files
Giving group "tripgrp" group ownership of the TRIP files

The user who should run TRIPserver is:
    root
/opt/trip/sys/v820/bin /opt/trip/sys/v820
/opt/trip/sys/v820

Path to libtdbs.so added to ld.so.conf

Setting up symbolic link for TRIPsystem libraries:

Creating new TRIPsystem library link: /usr/local/trip/sys/lib ->
/opt/trip/sys/v820/lib

Distributed CONTROL template file is /opt/trip/sys/v820/sys/P_CONTROL.*
- copying to /var/lib/trip/ctl

Initializing new CONTROL database from template in /var/lib/trip/ctl

Creating environment definition scripts/config file
Generating new configuration file (tdbs.conf)

Make installation conf directory
Copying /opt/trip/sys/v820/TRIPrcs.template to
    /opt/trip/sys/v820/conf/tdbs.conf

Normal case - /usr/local/trip/sys/conf is a directory link
Creating link /usr/local/trip/sys/conf to:
    /opt/trip/sys/v820/conf

**** TRIP System Utility SETSPWD - Set TRIP System Password ****
      Version 8.2-0   02-May-2022 12:20:37

New password          :
Verification          :

Elapsed: 00:00:02

```

Password set for TRIP system

Service pctdbs(n) already configured in file /etc/services as 23567

umask is reset to old value: 0022

Configuring TRIP services to be managed by systemd

Created symlink /etc/systemd/system/multi-user.target.wants/tripd.service
→ /etc/systemd/system/tripd.service.

Service tripd successfully installed and started.

Use the systemctl command to start, stop, enable and disable tripd.

Created symlink /etc/systemd/system/multi-user.target.wants/tripnetd.service →
/etc/systemd/system/tripnetd.service.

Service tripnetd successfully installed and started.

Use the systemctl command to start, stop, enable and disable tripnetd.

***** End of installing TRIP v8.2-0 on Linux

Before using the new TRIP installation,
please log out and then log in again!

End of install script!

NB!

Please check the installation log file for error messages!

#

Upgrade installation

The following is an example of an upgrade installation of version 8.1-3 to version 8.2-0 of a TRIPsystem installed on RedHat Enterprise Linux 8 platform. This is the text displayed on the console/terminal:

#

v820/sbin/install

PLEASE NOTE!

When prompted for a response, entering a ? gives a short help text

Beginning installation of TRIP v8.2-0 on a Linux x86_64 platform.

Do you wish to continue? [Y|n|?]

***** Answer: default [= Y]

Continuing with the installation process

NB! Installation log file is:

/opt/trip/sys/v820/install.log

Checking for presence of prerequisite software ...

bash: 4.4.20

sed: 4.5

libstdc++: 8.5.0

gawk: 4.2.1

binutils: 2.30

procps-ng: 3.3.15

```
psmisc: 23.1
file: 5.33
libgcc: 8.5.0
dmidecode: 3.2
libc: 60.3
openssl: 1.1.1k
expat: 2.2.5
zlib: 1.2.11
```

Good news! All prerequisites are already installed.

umask is changed to 022

***** Install TRIP v8.2-0 on a Linux x86_64 platform

2022-05-02 13:38:33

```
Detected existing TRIP config file /usr/local/trip/sys/conf/tdbs.conf
Found /usr/lib/libtdbs.so
/usr/lib/libtdbs.so links to
/opt/trip/sys/v813/bin/libtdbs.so
```

```
The TRIP daemon (tripd) is running. Trying to stop it now!
The TRIP daemon (tripd) has been stopped.
The TRIP daemons can be installed for automatic start via systemd.
Do you wish to start the TRIP daemons via systemd? [y|N|?] y
```

***** Answer: y

Found old TRIP configuration file: /opt/trip/sys/v813/conf/tdbs.conf

```
Found existing TRIP kernel library in /usr/lib
linking to: /opt/trip/sys/v813/bin/libtdbs.so
```

Is this a [N]ew installation or an [U]pgrade? [U]

***** Answer: default [= U]

Proceeding with an upgrade of TRIP

Which user should run the TRIP daemons? [root]

***** Answer: default [= root]

Which group should run the TRIP daemons? [tripgrp]

***** Answer: default [= tripgrp]

```
The TRIP language is set to ENG
The TRIP character set is set to LA1
The default location for TRIP databases is set to /usr/local/trip/data
```

```
Setting TRIP file protections
Setting TRIP file ownership to:
"root:tripgrp" (copied from previous installation)
Giving user "root" ownership of the TRIP files
Giving group "tripgrp" group ownership of the TRIP files
```

```
The user who should run TRIPserver is:
root (copied from previous installation)
/opt/trip/sys/v820/bin /opt/trip/sys/v820
/opt/trip/sys/v820
```

Path to libtdbs.so added to ld.so.conf

Setting up symbolic link for TRIPsystem libraries:

```
Current TRIPsystem library link removed from /usr/local/trip/sys
Creating new TRIPsystem library link: /usr/local/trip/sys/lib ->
/opt/trip/sys/v820/lib
```

```
Distributed CONTROL template file is /opt/trip/sys/v820/sys/P_CONTROL.*
- copying to /var/lib/trip/ctl
Distributed CONTROL template file is /var/lib/trip/ctl/P_CONTROL.*
Copying /opt/trip/sys/v813/sys/CONTROL.* to
/var/lib/trip/ctl/CONTROL.*
The CONTROL files should be in /var/lib/trip/ctl
Verifying... ok
```

```
The P_CONTROL files should be in /var/lib/trip/ctl
Verifying... ok
```

```
Pruning old CONTROL database backups if there are more than 10
```

```
Backing up the CONTROL database files with suffix 20220502133842
'/var/lib/trip/ctl/CONTROL.BAF' -> '/var/lib/trip/ctl/CONTROL.BAF-
20220502133842'
'/var/lib/trip/ctl/CONTROL.BIF' -> '/var/lib/trip/ctl/CONTROL.BIF-
20220502133842'
'/var/lib/trip/ctl/CONTROL.VIF' -> '/var/lib/trip/ctl/CONTROL.VIF-
20220502133842'
```

```
**** TRIP System Utility PACKIT - Compress a BAF file ****
Version 8.2-0 02-May-2022 13:38:42
```

```
Database : CONTROL
Character set for new BAF file : As for old BAF
*****
*** This database contains non-indexed modified records. ***
*** If you continue, make sure that the database is ***
*** completely reindexed afterwards. ***
*****
Do you want to continue PACKIT? (N): Y
Current BAF file : TDBS$CTL:CONTROL.BAF
New BAF file specification :
/var/lib/trip/log/sys/__CONTROL_PACKIT__.BAF
```

```
Number of records processed: 455
```

```
Elapsed: 00:00:00
```

```
**** TRIP System Utility INDEX - Create/Update BIF/VIF Files ****
Version 8.2-0 02-May-2022 13:38:42
```

```
Data base : CONTROL
```

```
----- Index files will be regenerated -----
```

```
STO file name :
/var/lib/trip/log/sys/CONTROL_222935.STO
BUT file name :
/var/lib/trip/log/sys/CONTROL_222935.BUT
```

```
----- Starting BIF scan phase. ----- 02-May-2022 13:38:42
```

```
Number of new main records: 455
Number of new paragraph records: 0
Number of modified main records: 0
Number of modified paragraph records: 0
```

```
Number of text occurrences: 121
Number of phrase occurrences: 9944
```

```

Number of number occurrences:      1535
Number of date occurrences:        910
Number of time occurrences:        910

```

```

Number of subfiles created:        0
Number of terms:                   1861
Number of sepx terms:              0
Number of overflow terms:          0
Number of hash code coincidences:  0

```

```

----- Starting BIF update phase. ----- 02-May-2022 13:38:42

```

```

----- Current Entry Block Width:: 5 ----- 02-May-2022 13:38:42

```

```

No. of terms added/updated:        1861

```

```

----- Starting VIF scan phase. ----- 02-May-2022 13:38:42

```

```

Number of scanned terms:           2531

```

```

Number of subfiles created:        0
Number of terms:                   3139
Number of overflow terms:          0
Number of hash code coincidences:  0

```

```

----- Starting VIF update phase. ----- 02-May-2022 13:38:42

```

```

----- Current Entry Block width:: 5 ----- 02-May-2022 13:38:42

```

```

----- Reorganizing the inverted file ----- 02-May-2022 13:38:42

```

```

----- New Entry Block Width:: 6 ----- 02-May-2022 13:38:42

```

```

No. of terms added/updated:        3139

```

```

----- Removing old BAF record versions ----- 02-May-2022 13:38:42

```

```

Exit 02-May-2022 13:38:42

```

```

Elapsed: 00:00:00

```

The CONTROL database was successfully re-indexed.
 Proceeding with the migration proper of the CONTROL database.

```

**** TRIP System Utility MODCON - Modify CONTROL records ****
      Version 8.2-0 02-May-2022 13:38:42

```

```

Processing system database designs...
Processing English forms/formats...
Processing Swedish forms/formats...
Processing Norwegian forms/formats...
Processing German forms/formats...
Processing Finnish forms/formats...
Processing Chinese forms/formats...

```

```

Elapsed: 00:00:01

```

The CONTROL database migration completed successfully.


```
Creating environment definition scripts/config file
Copying configuration file from /opt/trip/sys/v813/conf/tdbs.conf
'/opt/trip/sys/v813/conf/tdbs.conf' -> '/opt/trip/sys/v820/TRIPrcs.temp'
```

```
Make installation conf directory
Copying /opt/trip/sys/v820/TRIPrcs.template to
/opt/trip/sys/v820/conf/tdbs.conf
```

```
Normal case - /usr/local/trip/sys/conf is a directory link
Removing link from /usr/local/trip/sys/conf to:
/opt/trip/sys/v813/conf
Creating link /usr/local/trip/sys/conf to:
/opt/trip/sys/v820/conf
```

Service pctdbs(n) already configured in file /etc/services as 23567

umask is reset to old value: 0022

```
Configuring TRIP services to be managed by systemd
Created symlink /etc/systemd/system/multi-user.target.wants/tripd.service
→ /etc/systemd/system/tripd.service.
```

```
Service tripd successfully installed and started.
Use the systemctl command to start, stop, enable and disable tripd.
Created symlink /etc/systemd/system/multi-
user.target.wants/tripnetd.service →
/etc/systemd/system/tripnetd.service.
```

```
Service tripnetd successfully installed and started.
Use the systemctl command to start, stop, enable and disable tripnetd.
```

***** End of installing TRIP v8.2-0 on Linux

Before using the new TRIP installation,
please log out and then log in again!

End of install script!

NB!

Please check the installation log file for error messages!

#

Appendix B

TRIPserver init file

The environment variable **TBS_HOSTINI** defines the directory path for an optional client application activated server initialization file (Hostini). This file should then contain variables used to re-define the TRIP environment (TDBS_CTL, TDBS_SYS, etc.)

TRIP will first read variables from the TRIP config file and then read the Hostini file, replacing any Non-Privileged variables already read from the TRIP config file.

The TBS_HOSTINI variable is defined in the TRIP config file using the following syntax:
TBS_HOSTINI=<directory>

The variable name must include no spaces, be followed by an equals sign and have valid value. The line must also end with a carriage return. The default value for TBS_HOSTINI is /tmp.

Refer to client tools documentation for further information about how to use a Hostini file.

Appendix C:

Troubleshooting

There is a TRIP utility named `ckinst` that checks the status of the TRIPsystem installation. This utility, located in the `<TRIPpath>/sbin` directory, will report as warnings and errors items that should or must be addressed.

An example of a `ckinst` run on a healthy RedHat Enterprise Linux installation follows:

```
*****
*                TRIP ckinst script (upd: May 2 13:36)
*                Analyzing the current TRIP installation
*****

INFO: Current date is 2022-05-02 13:43:13
INFO: Running as user root
INFO: The Linux distribution family is rhel
INFO: Found /usr/local/trip/sys/conf
INFO: /usr/local/trip/sys/conf resolved to /opt/trip/sys/v820/conf
INFO: TDBS_HOME seems to be /opt/trip/sys/v820
INFO: Found tdb.conf under /opt/trip/sys/v820/conf
INFO: Checking for prerequisite software...
INFO: all prerequisite software packages are installed
INFO: Config says TDBS_EXE=/opt/trip/sys/v820/bin
INFO: TDBS_EXE location appears to be correct
INFO: /usr/local/trip/sys/lib/libtdbs.so links to
/opt/trip/sys/v820/lib/libtdbs.so
INFO: File /opt/trip/sys/v820/lib/libtripcrt_core.so is a link
INFO:      to /opt/trip/sys/v820/lib/libtripcrt_core.so.2.0.0
INFO: File /opt/trip/sys/v820/lib/libtripcrt_core++.so is a link
INFO:      to /opt/trip/sys/v820/lib/libtripcrt_core++.so.2.0.0
INFO: File /opt/trip/sys/v820/lib/libxerces-c.so is a link
INFO:      to /opt/trip/sys/v820/lib/libxerces-c-3.2.so
INFO: TDBS_SYS=/opt/trip/sys/v820/sys
INFO: Required system files are present in the TDBS_SYS directory
INFO: TDBS_CTL=/var/lib/trip/ctl
INFO: CONTROL database files found in the TDBS_CTL directory
INFO: Found 54 of 54 required files in TDBS_SYS
INFO: TRIP_DEMO=/opt/trip/sys/v820/demo
INFO: Found 18 of 18 required files in TRIP_DEMO
INFO: the tripnet port pctdb is defined in /etc/services as 23567
INFO: Neither of the obsoleted superservers (inetd or xinetd) is used
INFO: tripd is running as root
INFO: tripnetd is running as root
INFO: tbserver will be launched by tripnetd
INFO:      as /opt/trip/sys/v820/bin/tbserver
INFO: tbserver will be run as root
INFO: Found libtdbs.so via runtime linker in /usr/local/trip/sys/lib
INFO: TRIP version is 8.2-0:0

Found 0 error(s) and 0 potential problem(s) with your installation.
Your installation is OK!
```

If `ckinst` reports no problems and TRIP still does not work as expected, please contact TRIP support for help.