

# 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 Smaser AG (Smaser) are exclusively owned by Smaser.

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

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



# Table of Contents

<b>PREFACE.....</b>	<b>5</b>
<b>1. PREPARING FOR INSTALLATION .....</b>	<b>6</b>
1.1 PREREQUISITE HARDWARE AND SOFTWARE.....	6
1.2 PREREQUISITE DISK SPACE REQUIREMENTS .....	6
1.3 THE INSTALLATION PROCEDURE.....	6
<b>2. INSTALLING TRIPSYSTEM .....</b>	<b>7</b>
2.1 LOADING TRIP .....	7
2.2 STARTING THE INSTALLATION PROCESS .....	7
2.3 THE INSTALLATION MESSAGE.....	7
2.4 STOPPING THE TRIP DAEMON (TRIPD) .....	8
2.4.1 <i>Stopping the TRIPnet daemon (tripnetd)</i> .....	8
2.5 NEW OR UPGRADE INSTALLATION .....	8
2.6 COPYING THE CONTROL FILE.....	9
2.7 NEW INSTALLATION.....	9
2.7.1 <i>System start-up</i> .....	9
2.7.2 <i>Owner of the TRIP files</i> .....	9
2.7.3 <i>User who will run TRIPserver</i> .....	10
2.7.4 <i>Setting TRIP permissions</i> .....	10
2.7.5 <i>Default language</i> .....	10
2.8 UPGRADE INSTALLATION .....	11
2.8.1 <i>Owner of the TRIP files</i> .....	11
2.8.2 <i>Setting TRIP permissions</i> .....	11
2.8.3 <i>User who will run TRIPserver</i> .....	11
2.9 FILE SETTINGS.....	11
2.10 SETTING UP LIBRARY LINK .....	11
2.11 XPI INSTALLATION CHECK .....	11
2.12 3RD PARTY INSTALLATION .....	11
2.13 LDAP INSTALLATION CHECK.....	11
2.14 NEW INSTALLATION.....	12
2.14.1 <i>Creating the CONTROL file</i> .....	12
2.14.2 <i>Setting the TRIP system password</i> .....	12
2.14.3 <i>Setting the TRIPdaemon password</i> .....	12
2.15 UPGRADE INSTALLATION .....	12
2.15.1 <i>Upgrading the CONTROL file</i> .....	12
2.15.2 <i>Upgrading DEBIT.LOG, PRC, PRN and TRM files</i> .....	13
2.15.3 <i>Setting up the TRIP daemon</i> .....	13
2.16 TRIPSERVER INSTALLATION .....	14
2.16.1 <i>The TRIPnet daemon</i> .....	14
2.17 COMPLETION MESSAGES.....	14
2.18 ERROR MESSAGES .....	15
2.19 CHECKING THE INSTALLATION.....	15
<b>3. POST INSTALLATION REQUIREMENTS FOR TRIP .....</b>	<b>16</b>
3.1 STARTING THE SYSTEM .....	16
3.2 UPGRADING OLD DATABASE INDEX FILES .....	16
3.3 LOG FILES AND TEMPORARY FILES .....	16
3.4 TRIPSERVER .....	16
<b>4. THE TRIPDAEMON .....</b>	<b>17</b>
4.1 AN OVERVIEW OF THE TRIPDAEMON .....	17
4.2 TRIPDAEMON OPTIONS.....	17
4.2.1 <i>Summary of options</i> .....	17
4.2.2 <i>Starting the TRIPdaemon</i> .....	18
4.2.3 <i>Changing the TRIPdaemon password ( -a )</i> .....	18



4.2.4 Current information ( -d file ) .....	19
4.2.5 List current clients ( -c ) .....	19
4.2.6 Stopping the TRIPdaemon ( -k ) .....	19
4.2.7 TRIPdaemon logfile ( -l ) .....	20
4.2.8 Locked Record ( -o ) .....	20
4.2.9 Purge Clients (-P) .....	21
4.2.10 Current Queues ( -q ) .....	21
4.2.11 Restart daemon ( -r ) .....	21
4.2.12 TRIPdaemon Statistics ( -s ) .....	22
4.2.13 Log file Verbosity Level ( -vn ) .....	22
<b>5. REMOVING A TRIP INSTALLATION .....</b>	<b>23</b>
5.1 ROLLING BACK A TRIP INSTALLATION .....	23
<b>6. MANAGING TRIP IN A UNIX ENVIRONMENT .....</b>	<b>24</b>
6.1 PATH DEFINITION .....	24
6.2 FILE PROTECTION .....	24
6.3 ENVIRONMENT NAME SPACES .....	25
<b>APPENDIX A .....</b>	<b>27</b>
SAMPLE INSTALLATIONS .....	27
<i>New Installation</i> .....	27
<i>Upgrade installation</i> .....	29
<b>APPENDIX B .....</b>	<b>35</b>
TRIPSERVER INIT FILE .....	35
<b>APPENDIX C: .....</b>	<b>36</b>
TROUBLESHOOTING FOR THE TRIPSERVER (TBSERVER) .....	36



## 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 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 in Figure 1.

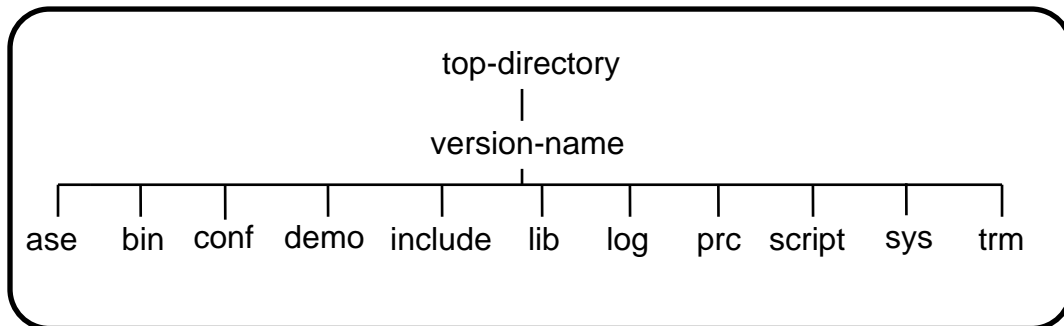


Figure 1: The TRIP directory structure.

In Figure 1, 'version-name' is defined by the installation procedure as the maintenance release number of the TRIP system being installed, for instance 'v721' for version v7.2-1.

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_linux_v721.tar
```

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

TRIPpath = <top-directory>/<version-name> (used later in this Installation Guide)

### 2.2 Starting the installation process

Still in <top-directory> (in Figure 1) you begin the installation procedure by entering:

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

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

```
v701/script/install
```

### 2.3 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, do you wish to continue [Y]?
```

If you accept the default answer (by hitting RETURN), or type an affirmative response, 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*.

## 2.4 Stopping the TRIP daemon (tripd)

The installation requires that the TRIP daemon is stopped. If it is running it will be stopped automatically by the installation procedure. Then you will get a prompt asking if you want the TRIP daemon started automatically at the end of the installation:

```
Do you want the TRIP daemon (tripd) started at end of installation? [Y]
```

The default for this question is Y if the TRIP daemon is already running, N if it is 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.

### 2.4.1 Stopping the TRIPnet daemon (tripnetd)

The installation requires that a running TRIPnet daemon is stopped. It will be stopped automatically by the installation procedure. Then you will get a prompt asking if you want the TRIPnet daemon started automatically at the end of the installation:

```
Do you want the TRIPnet daemon (tripnetd) started at end of installation? [N]
```

The default for this question is Y if the TRIPnet daemon is already running, N if it is not.

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 net/web services will not be available until the TRIPnet daemon has been started when it is used as an Internet services daemon instead of inetd or xinetd.

## 2.5 New or upgrade installation

As stated previously, 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
- 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

The next prompt will be:





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

The default answer to this prompt is based on the existence of a TRIP config file.

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.

## 2.6 Copying the CONTROL file

The name of the distributed control file is P\_CONTROL and depending on installation type, one of the following will be true:

- If this is a new installation, P\_CONTROL will be copied to CONTROL to become the control file for the installation.
- If it is an upgrade installation, the old CONTROL file will be copied to the new installation area.
- For a patch upgrade (i.e. when the version number contains a ":" ), no copying of control files will take place.

The CONTROL file stores control data for the databases, users, procedures, forms and formats. Therefore, having a copy of the original file is strongly recommended.

## 2.7 New installation

### 2.7.1 System start-up

You have the option of installing TRIP locally, or system wide. The difference between the options is that system wide start-up will install the TRIP config file in the <TRIPpath>/conf directory. Otherwise, the TRIP config file remains local in the <TRIPpath> directory. You will be prompted:

```
Would you like TRIP installed for [S]ystem wide startup or [L]ocal startup?
[S]
```

The default is for system wide start-up. If you select local start-up, you will be able to change to system wide start-up at a later date. Refer to Chapter 6 for further information.

### 2.7.2 Owner of the TRIP files

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. Refer to Chapter 6 for further information.

### 2.7.3 User who will run TRIPserver

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).

Even if any user can run the TRIPserver it is strongly recommended to choose root as that user. Refer to Chapter 6 for further information.

### 2.7.4 Setting TRIP permissions

You now have the option to set the group s-bit for the TRIP executables. By setting this option, TRIP will manage the access rights to TRIP databases. If you decline to set the group s-bit for the executables, the system manager is responsible to ensure that TRIP users have all necessary permissions to read/write of directories and databases.

```
Do you wish to have the s-bit set for the TRIP executables [Y]?
```

The default option [Y] will set the s-bit for all executables. If you enter an invalid response, you will be prompted until 'y' or 'n' is entered. Refer to Chapter 6 for further information.

### 2.7.5 Default language

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. During the installation, the system default language is specified:

```
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`.

## 2.8 Upgrade installation

### 2.8.1 Owner of the TRIP files

The owner of the TRIP files (user and group) will be set to the same as in the current installation.

### 2.8.2 Setting TRIP permissions

The group s-bit for the TRIP executables will be set for those executables that have the s-bit set in the current installation.

### 2.8.3 User who will run TRIPserver

The user who will run the TRIPserver will be set to the same as in the current installation.

## 2.9 File settings

The actual setting of file protections, s-bits and file ownership (user and group) will be done now.

## 2.10 Setting up library link

A logical link will now be created from `/usr/lib` to the installed TRIP runtime library.

## 2.11 XPI installation check

Checks if XPI is present in the current installation.

TRIPxpi is the server side component used by TRIPmgr, the Windows TRIP Administration Program as well as all the new TRIPsdk products TRIPnxp, TRIPjxp and TRIPaxp.

For further information, refer to the documentation available with each of these products.

## 2.12 3rd party installation

If present for the current platform, 3rd party software will be installed by running a separate script, *inst3rdp*, which is located in the script directory of the installation.

Included in this software is ICU, which is a package for handling Unicode, Xerces, which is a collection of software libraries for parsing, validating, serializing and manipulating XML and Mozilla LDAP

## 2.13 LDAP installation check

Checks if LDAP is present in the current installation.



For further information, refer to the file instldap.txt in the <TRIPpath>/script directory or the TRIP6\_New\_Feature document.

## 2.14 New installation

### 2.14.1 Creating the CONTROL file

The name of the distributed control file is P\_CONTROL and the following message will appear (example):

```
Distributed CONTROL file is <version 7.2-0>/P_CONTROL.*
```

P\_CONTROL will be copied to CONTROL to become the control file for the installation.

### 2.14.2 Setting the TRIP system password

You will be prompted to provide and verify the TRIP SYSTEM user's password:

```
**** TRIP System Utility SETSPWD - Set TRIP System Password ****
Version 7.2-0    16-Mar-2018 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.

### 2.14.3 Setting the TRIPdaemon password

To start, and to subsequently use, the TRIPdaemon, you must assign a password to the daemon. If this is a new installation of TRIP, you will see the following prompt:

```
**** TRIP System Utility SETDPWD - Set TRIPdaemon Password ****
Version 7.2-0    16-Mar-2018 08:49:19

New Password          :
Verification          :
```

Neither the password nor the verification will be echoed to the screen. If the verification does not match the supplied password, the prompts will loop. The program will loop until the new password and the verification match.

## 2.15 Upgrade installation

### 2.15.1 Upgrading the CONTROL file

The name of the distributed control file is P\_CONTROL and the following message will appear (example):

```
Distributed CONTROL file is <version 6.0-1>/P_CONTROL.*
```

When upgrading an existing installation, the installation process will try to find the current system wide CONTROL file. If the file is found, you will see the following message (example):

```
Copying <original>/CONTROL.* to <version 6.0-1>/CONTROL.*
```



Here <original> refers to the CONTROL file that currently resides on the system and this file will be copied to the installation area. This CONTROL file then needs to be updated to accommodate enhancements to be found in the new version.

Before running utility program MODCON, the CONTROL file will be completely re-indexed to make sure that MODCON works properly. The following message will appear when indexing begins:

```
Indexing Control
```

When the indexing is completed this message will appear:

```
Modifying CONTROL Records for new version
```

Whereby MODCON is activated and the message appears:

```
**** TRIP System Utility MODCON - Modify CONTROL Records ****
      Version 6.0-1 10-Dec-2008 10:11:12

Processing system data base designs
Processing English forms/formats ...
Processing Swedish forms/formats ...
Processing Norwegian forms/formats ...
Processing German forms/formats ...
```

This message shows the installation procedure upgrading your existing CONTROL file with new formats and database definitions required for the new version of TRIP to function correctly.

If any errors occur during the above process, refer to the installation script for more information.

### 2.15.2 Upgrading DEBIT.LOG, PRC, PRN and TRM files

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>
```

If any terminal files (.TRM) are used with the current version of TRIP, these files will be copied to the new trm directory provided TDBS\_TRM is defined in the TRIP config file. This will produce the message:

```
Copying TRM files from <current-version> to <new-version>
```

### 2.15.3 Setting up the TRIP daemon

If the password file (TRIPD.DAT) for the TRIP daemon exists in the current <TRIPpath>/bin directory, then that file is copied to the new version directory. If the password file does not exist, then the password for the daemon must be set. You will see the following:

```
**** TRIP System Utility SETDPWD - Set TRIPdaemon Password ****
      Version 6.0-1 10-Dec-2008 10:11:12
```



```
New Password      :
Verification      :
```

Neither the password nor the verification will be echoed to the screen. If the verification does not match the supplied password, the prompts will loop. The program will loop until the new password and the verification agree.

## 2.16 TRIPserver installation

TRIPserver is automatically installed as part of the TRIP installation process.

TRIPserver will be installed by running a separate script, *insttbs*, which is located in the script directory of the installation.

Different platforms use different daemons for net (Internet) connections. Linux platforms use *xinetd*, other UNIX platforms usually use *inetd* and Sun platforms from Solaris 10 upwards use yet another type that is maintained via the *inetadm* command.

It is also possible to use the TRIPnet daemon as an Internet Services daemon instead of those mentioned above.

The *insttbs* installation procedure tries these three types in the order *inetadm*, *xinetd*, *inetd*, unless you have required the TRIPnet daemon to be started, when this one will be used instead.

### 2.16.1 The TRIPnet daemon

On Windows platforms, the TRIPserver program *tbserver* is started by *tripnetd*. This service is listening for client session requests and launches an instance of *tbserver* to take care of any session requested.

This daemon can now also be used on Unix platforms in the same way, replacing the daemons mentioned above, *inetadm/inetd* and *xinetd*.

The TRIPnet daemon works in the same way as the other daemons, launching *tbserver* to service the client requests, but does not have all the functionality found in these, regarding e.g. defence against malicious (DOS) attacks.

Invoking the daemon with the help option gives this output (*tripnetd help*):

USAGE: *tripnetd* [command]

COMMANDS:

```
help      Display this help text
console   Run tripnetd with console logging
start     Start the tripnetd daemon process (default command)
```

Invoking *tripnetd* without a command starts the TRIPnet daemon normally.

NOTE: The TRIPnet daemon must be stopped by sending the SIGTERM signal to it.

## 2.17 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.



## 2.18 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 (in 2.1). You should, as the actual message indicates, attempt to correct the reported problem(s) before restarting the installation.

## 2.19 Checking the installation

You can use the new script 'chkinst', that can be found in the 'script' directory of the installation, to check if your installation is correctly setup. You should run this script as root to get the most correct information, and the script checks the following:

- The presence of a configuration file
- The presence of /usr/lib/libtdbs.so
- That the configuration file is tdbs.conf or .TRIPrcs
- That /usr/local/trip/sys/conf links to \$TDBS\_EXE/./conf
- That /usr/lib/libtdbs.so links to \$TDBS\_EXE/libtdbs.so
- That the dependencies of the TRIP binaries are fulfilled
- The presence of 3rd-party, stemming and XPI support
- The presence of the CONTROL database
- The presence of a DEBIT log file
- If the tbserver is set up for access via inetd
- If the tbserver is set up for access via xinetd
- If tripd is running
- If tripnetd is running and potentially in conflict with inetd/xinetd
- That required files in the installation directories are present
- The version of the installation.



## 3. Post installation requirements for TRIP

### 3.1 Starting the system

You must now decide on whether you want the TRIP system to be started system wide, or simply process wide. During the installation process, you either selected system wide or local (process wide) installation.

A process wide installation requires a copy of the TRIP config file to be placed in the home directory of every TRIP user or that appropriate environment variables are included in the user start-up scripts. Refer to Chapter 6 for further information.

### 3.2 Upgrading old database index files

If you have upgraded a previous TRIP installation older than version 4.1, please read this paragraph!

To take advantage of some of the new features in TRIP, you must ensure that your databases are re-indexed. This is done by first running the *bafini* utility program and then indexing the database. This must be done for every database where you wish to use some of the new features of TRIP that require a re-index.

### 3.3 Log files and temporary files

It is recommended that the **TDBS\_LOG** variable in the TRIP config file is setup to point at a directory where all log files created by TRIP are stored. A new installation will add this variable with the value <TRIPpath>/log.

It is also recommended that the **TDBS\_SCRATCH** variable in the TRIP config file is setup to point at a directory where all temporary files created/used by TRIP are stored.

It is also recommended that the **TDBS\_SIF** variable in the TRIP config file is setup to point at a directory where the temporary session files created/used by TRIP are stored.

### 3.4 TRIPserver

If **TDBS\_SCRATCH** is not defined, and the TRIPserver process *tbserver* is started under 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, TRIPserver is likely to fail.

NB: 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 TRIPserver has been correctly installed and that your network is enabled for TRIP.





## 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 TRIPdaemon options

Whether you opt to start the TRIP system using the system wide start-up mechanism or the process start-up mechanism, you must start the TRIPdaemon. If you do not start the daemon, 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 file
-c	list current users
-kn	kill the daemon in n minutes
-lfile	set the logfile to file
-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 n

#### 4.2.2 Starting the TRIPdaemon

To start the TRIP daemon, enter the following:

```
<TRIPbin>/tripd
```

```
TRIPbin = <TRIPpath>/bin
```

The only option that may be specified when starting the daemon is *-l file* (refer to 4.2.7).

If you would like the daemon started up on boot up of the system, place the call 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 root. Otherwise, you will experience unpredictable results.

If you receive errors during the TRIP daemon boot process that say:

```
Cannot open $TDBS_EXE/TRIPD.DAT
Run setdpwd before starting the daemon
```

check to see if the TRIP config file exists in the root directory (for system wide startup).

#### 4.2.3 Changing the TRIPdaemon password ( -a )

If you suspect that the security of the TRIPdaemon password has been breached, you can change the daemon password with the command:

```
<TRIPbin>/tripd -a
```

which will prompt you as follows :

```
* Old password :
* New password :
* Verify password :
```

If the "verify" does not match the "new", these prompts will loop. Once they do, however, a request will be sent to the detached TRIPdaemon to change the password. If the password supplied as the "old password" was correct, the password will be changed, both in memory and in the password file, TRIPD.DAT located in the <TRIPpath>/bin directory. If the old password entered was invalid, then the message:

```
Invalid password - Password not changed
```

will appear.



#### 4.2.4 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:

```
<TRIPbin>/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
*****
```

#### 4.2.5 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:

```
<TRIPbin>/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.

#### 4.2.6 Stopping the TRIPdaemon ( -k )

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

```
<TRIPbin>/tripd -k
```

You will be prompted to enter the TRIP daemon password:

```
* TRIPdaemon password:
```

The password you enter will not be echoed to the screen. If the password you enter is incorrect, you will receive the message:

```
Invalid password
```



and be returned to the system prompt. The daemon will NOT be shutdown. If you enter a correct password, you will then be prompted for a delay time before shutdown:

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

If you would like to simplify the keystrokes, you may enter the password on the system line. However, note that the password is not hidden and anyone may observe what you are typing. For example, type:

```
<TRIPbin>/tripd -k -p thispassword
```

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

```
<TRIPbin>/tripd -k2
```

#### 4.2.7 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:

```
<TRIPbin>/tripd -llogfile.txt
```

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

```
[12/09/08 18:44:16] TRIPdaemon v6.0-1
[12/09/08 18:44:16] -----
[12/09/08 18:48:59] dump_information: Dumped information to
'/tmp/EYaABFmAAA'.
[12/09/08 18:48:59] TRIPdaemon v6.0-1
[12/09/08 18:48:59] -----
[12/09/08 20:54:11] TRIPdaemon v6.0-1
[12/09/08 20:54:11] -----
[12/10/08 16:13:54] TRIPdaemon v6.0-1
[12/10/08 16:13:54] -----
[12/10/08 16:13:54] dump_information: Dumped information to
'/tmp/EmdABFmAAA'.
[12/10/08 16:14:07] TRIPdaemon v6.0-1
[12/10/08 16:14:30] -----
[12/10/08 18:50:58] dump_information: Dumped information to
'/tmp/EYNMdasATM'.
```

#### 4.2.8 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.



#### 4.2.9 Purge Clients (-P)

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

```
<TRIPbin>/tripd -P
```

You will be prompted to enter the TRIP daemon password:

```
* TRIPdaemon password:
```

The password you enter will not be echoed to the screen. If the password you enter is incorrect, you will receive the message:

```
Invalid password
```

and be returned to the system prompt. If you enter a correct daemon password, then the clients are purged and you are returned to the system prompt. No messages are given.

#### 4.2.10 Current Queues (-q)

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

```
<TRIPbin>/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.

#### 4.2.11 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:

```
<TRIPbin>/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. When you issue the above command, you will be prompted for the daemon password:

```
* TRIPdaemon password:
```

The password you enter will not be echoed to the screen. If the password you enter is incorrect, you will receive the message:

```
Invalid password
```

and be returned to the system prompt. If you enter a correct daemon password, then the daemon will be restarted. There will be no messages displayed.



#### 4.2.12 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:

```
<TRIPbin>/tripd -s
```

would result in a display as such:

```
TRIPdaemon v6.0-1, started 12/10/08 12:14:30

JOBS          QUEUED    FORKED    EXITED    KILLED    FORKERR
              20         20        20         0          0

QUEUES        CREATED    EXPIRED    KILLED
              7          7          0

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

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

#### 4.2.13 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 are:

- 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:

```
<TRIPbin>/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:

```
<TRIPbin>/tripd -v65
```

You can change the log file of a running daemon. To change the verbosity level, you will be prompted to enter the TRIP daemon password:

```
* TRIPdaemon password:
```

If you enter an invalid password, you will see the message:

```
Invalid password
```

and be returned to the system prompt.



## 5. Removing a TRIP installation

There is no automated 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).

The reason for this is that s-bit group privilege is granted to the TRIP executables (mode 2755) to allow them to grant the calling user temporary membership of a named group (the one specified by the manager during installation).





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

```
-rwxr-sr-x      root      tripgrp      .. baffit
-rwxr-xr-x      root      tripgrp      .. lixit
-rwxr-xr-x      root      tripgrp      .. tdbstrt1.o
```

i.e. those programs which require 'privilege' (baffit, baffre, bafini, sciffit, tbserver, tdbglbupd, tdbprint and trip) are set with s-bit group privilege (note the 's' bit on group access), whilst all other files are simply execute protected (including the kernel library which gets no special privilege of its own).

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.

There is, in fact, no requirement to have any 'other' access at all, as TRIP will always access these files in 'privileged' mode (i.e. following a setegid() call). Nor is there any requirement to have either the TRIP files or the executables owned by 'root'.

Also note that in order for this security mechanism to function correctly, the group specified for ownership should not have any members listed in /etc/group (or equivalent).

## 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=/home/accounting
TDBS_ACCFLG=8

[NonPrivileged]
TDBS_CTL=/home/trip/v601/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=/home/trip/v701/ctl
[Group:developers]
TDBS_CTL=/home/trip/v701/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

The install.log file now contains the same text as in both cases below, from the line:

```
'umask is changed to 022'                                # First line logged!
```

to the line:

```
'End of install script!'                                # Last line logged!
```

### New Installation

The following is an example of a new installation of a system-wide version 7.0-2 of a TRIPsystem installed on a 32-bit Sun Solaris platform; this is the text displayed on the console/terminal:

```
#
# v702/script/install
PLEASE NOTE!
  when prompted for a response, entering a ? gives a short help text

Beginning installation of TRIP v7.0-2 on a SunOS platform.
Do you wish to continue? [Y]

Continuing with the installation process

NB! Installation log file is:
    /bigdisk/trip/v702/install.log

umask is changed to 022                                # First line logged!
***** Install TRIP v7.0-2 on SunOS *****

2014-02-17 10:26:51

The TRIP daemon (tripd) was not running.

Do you want the TRIP daemon (tripd) started at end of installation? [N]
The TRIPnet daemon (tripnetd) was not running.

Do you want the TRIPnet daemon (tripnetd) started at end of installation?
[N]
Is this a [N]ew installation or an [U]pgrade? [N]
Proceeding with a new installation of TRIP

which user should own the TRIP files? [root]
which group should own the TRIP files?
which user should run the TRIPserver (tbserver)? [root]
would you like TRIP installed for [S]ystem wide startup or [L]ocal
startup? [S]
TRIP will be installed for SYSTEM WIDE use

Do you wish to have the Set-Group-Id bit set for the TRIP executables?
[Y] The S
et-Group-Id bit is set for the TRIP executables

what is the default language when running TRIP?

[E]nglish / [S]wedish / [G]erman / [F]innish / [C]hinese / [N]orwegian
[E]:
The TRIP language selected was ENG
```



```
( Setting the character set to LA1 )

Setting TRIP file protections
chmod: WARNING: can't access log/*
chmod: WARNING: can't access prc/*
Setting TRIP file ownership to:
    "root:sys"
Giving user "root" ownership of the TRIP files
Giving group "sys" group ownership of the TRIP files

The user who should run TRIPserver is:
    root

Setting TRIP file permissions
Setting group s-bit for:
    bin/bafini
    bin/glbupdit
    bin/index
    bin/load
    bin/printit
    bin/tbserver
    bin/trip

Creating link from /usr/lib/libtdbs.so to:
    /bigdisk/trip/v702/bin/libtdbs.so
Creating link from /usr/lib/64/libtdbs.so to:
    /bigdisk/trip/v702/bin/libtdbs.so

TRIPxpi is present in the current installation

calling /bigdisk/trip/v702/script/inst3rdp with parameters:
    /bigdisk/trip/v702 /usr/local/trip/sys

***** 3rd-party installation script *****

2014-02-17 10:27:16

Script called with:
    Param. 1 TPATH = /bigdisk/trip/v702
    Param. 2 SYSDIR = /usr/local/trip/sys

PATH to installed 3rd-party libraries: /bigdisk/trip/v702/lib

Setting up symbolic link for 3rd-party dependencies:

Creating link directory: /usr/local/trip/sys
Creating new 3rd-party link: /usr/local/trip/sys/lib ->
    /bigdisk/trip/v702/lib

***** 3rd-party Installation finished *****

3rd-party installation successfully finished

LDAP is present in the current installation

Distributed CONTROL file is /bigdisk/trip/v702/sys/P_CONTROL.*
- copying to /bigdisk/trip/v702/sys/CONTROL.*

Creating environment definition scripts

Copying /bigdisk/trip/v702/TRIPrcs.template to
    /bigdisk/trip/v702/conf/tdbs.conf

Normal case - /usr/local/trip/sys/conf is a directory link
Creating link /usr/local/trip/sys/conf to:
    /bigdisk/trip/v702/conf

calling /bigdisk/trip/v702/script/insttbs with parameters:
```



```

        /bigdisk/trip/v702 root SunOS N

***** TRIPserver installation script *****

2014-02-17 10:27:17

Script called with:
  Param. 1 TPATH = /bigdisk/trip/v702
  Param. 2 TBUSER = root
  Param. 3 OSNAME = SunOS
  Param. 4 START_TND = N

File /etc/services is copied to /etc/cp.services.1 for backup
Service pctdbs is added last in file /etc/services :
pctdbs          23457/tcp    # TRIP Network Service ; 2014-02-17 10:27:17

Copying file /etc/inetd.conf to /etc/cp.inetd.conf.2 for backup
Adding service "pctdbs" last in file /etc/inetd.conf :
pctdbs stream tcp nowait root /bigdisk/trip/v702/bin/tbserver
tbserver

Restarting inetd

***** TRIPserver installation finished *****

TRIPserver installation successfully finished
The TRIPnet Daemon (tripnetd) should NOT be started!

umask is reset to old value: 0002

Password set for TRIP system

Password set for the TRIP daemon

The TRIP daemon (tripd) is not running. Trying to start it now!

The TRIP daemon (tripd) has been started.
  root 3622      1  0 10:29:00 ?          0:00
/bigdisk/trip/v702/bin/tripd

***** The TRIPnet daemon (tripnetd) was not started.
        If you intend to use it you must start it manually
        to make TRIP available to the net/web users.

***** End of installing TRIP v7.0-2 on SunOS *****

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

End of install script!                                # Last line logged!

NB!
Please check the installation log file for error messages!"

#

```

### Upgrade installation

The following is an example of an upgrade installation of a system-wide version 7.0-1 to version 7.0-2 of a TRIPsystem installed on a 32-bit Sun Solaris platform; this is the text displayed on the console/terminal:

```

#
# v702/script/install
PLEASE NOTE!

```



```

When prompted for a response, entering a ? gives a short help text
Beginning installation of TRIP v7.0-2 on a SunOS platform.
Do you wish to continue? [Y]

Continuing with the installation process

NB! Installation log file is:
    /bigdisk/trip/v702/install.log

umask is changed to 022

***** Install TRIP v7.0-2 on SunOS *****

2014-02-17 12:53:52

Detected existing TRIP config file /usr/local/trip/sys/conf/tdbs.conf
Found /usr/lib/libtdbs.so
/usr/lib/libtdbs.so links to
    /bigdisk/trip/v701/bin/libtdbs.so

The TRIP daemon (tripd) is running. Trying to stop it now!
The TRIP daemon (tripd) has been stopped.

Do you want the TRIP daemon (tripd) started at end of installation? [Y]

The TRIPnet daemon (tripnetd) was not running.

Do you want the TRIPnet daemon (tripnetd) started at end of installation?
[N]

Found old TRIPrcs file: /bigdisk/trip/v701/conf/tdbs.conf

Found old link in /usr/lib to the TRIP kernel library:
    /bigdisk/trip/v701/bin/libtdbs.so

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

Proceeding with an upgrade of TRIP

The TRIP language is set to ENG
The TRIP character set is set to MUL

Setting TRIP file protections
chmod: WARNING: can't access log/*
chmod: WARNING: can't access prc/*
Setting TRIP file ownership to:
    "root:sys" (copied from previous installation)
Giving user "root" ownership of the TRIP files
Giving group "sys" group ownership of the TRIP files

The user who should run TRIPserver is:
    _PREV_TB_USER_ (copied from previous installation)
Please refer to (x)inetd listing later in the log file

Setting TRIP file permissions
Setting group s-bit for:
    bin/bafini
    bin/glbupdit
    bin/index
    bin/load
    bin/printit

    bin/tbserver

    bin/trip
    
```



```

Creating link from /usr/lib/libtdbs.so to:
    /bigdisk/trip/v702/bin/libtdbs.so
Creating link from /usr/lib/64/libtdbs.so to:
    /bigdisk/trip/v702/bin/libtdbs.so

TRIPxpi is present in the current installation

Calling /bigdisk/trip/v702/script/inst3rdp with parameters:
    /bigdisk/trip/v702 /usr/local/trip/sys

***** 3rd-party installation script *****

2014-02-17 12:54:01

Script called with:
    Param. 1 TPATH = /bigdisk/trip/v702
    Param. 2 SYSDIR = /usr/local/trip/sys

PATH to installed 3rd-party libraries: /bigdisk/trip/v702/lib

Setting up symbolic link for 3rd-party dependencies:

Current 3rd-party link removed from /usr/local/trip/sys
Creating new 3rd-party link: /usr/local/trip/sys/lib ->
    /bigdisk/trip/v702/lib

***** 3rd-party Installation finished *****

3rd-party installation successfully finished

LDAP is present in the current installation

Distributed CONTROL file is /bigdisk/trip/v702/sys/P_CONTROL.*
Copying /bigdisk/trip/v701/sys/CONTROL.* to
    /bigdisk/trip/v702/sys/CONTROL.*

Indexing CONTROL

**** TRIP System Utility INDEX - Create/Update BIF/VIF Files ****
    Version 7.0-2    17-Feb-2014 12:54:02

Data base                      : CONTROL

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

STO file name                  :
/bigdisk/trip/v702/CONTROL_16086.STO
BUT file name                  :
/bigdisk/trip/v702/CONTROL_16086.BUT

----- Starting SCAN phase for BIF. ----- 17-Feb-2014 12:54:03

Processed 1024 (of 3444) New Records ... 17-Feb-2014 12:54:04
Processed 2048 (of 3444) New Records ... 17-Feb-2014 12:54:05
Processed 3072 (of 3444) New Records ... 17-Feb-2014 12:54:06

Number of new main records:      3164
Number of new paragraph records: 0
Number of modified main records: 0
Number of modified paragraph records: 0

Number of text occurrences:      39348
Number of phrase occurrences:    70655
Number of number occurrences:    9485
Number of date occurrences:      6328

```



```

Number of time occurrences:          6328
Number of subfiles created:          0
Number of terms:                    12424
Number of overflow terms:            18
Number of hash code coincidences:    0

----- Starting BIF update phase. ----- 17-Feb-2014 12:54:06
----- Current Entry Block width:: 8 ----- 17-Feb-2014 12:54:06
No. of terms added/updated:          12424
----- Starting SCAN phase for VIF. ----- 17-Feb-2014 12:54:06
Processed 16384 Terms ... 17-Feb-2014 12:54:07
Number of scanned terms:              17709
Number of subfiles created:            0
Number of terms:                      13246
Number of overflow terms:              17
Number of hash code coincidences:      0

----- Starting VIF update phase. ----- 17-Feb-2014 12:54:07
----- Current Entry Block width:: 8 ----- 17-Feb-2014 12:54:07
No. of terms added/updated:           13246

----- Removing old BAF record versions ----- 17-Feb-2014 12:54:07
Exit 17-Feb-2014 12:54:08
Elapsed: 00:00:06

Modifying CONTROL records for new version

**** TRIP System Utility MODCON - Modify CONTROL records ****
      Version 7.0-2   17-Feb-2014 12:54:08

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: 00:00:29

/bigdisk/trip/v701/sys/DEBIT.LOG exists,
  copying the file to /bigdisk/trip/v702/sys

Copying PRN files from /bigdisk/trip/v701/prc to
                        /bigdisk/trip/v702/prc

Copying old TRIPdaemon password file

Creating environment definition scripts

Copying /bigdisk/trip/v702/TRIPrcs.template to

```





```
/bigdisk/trip/v702/conf/tdbs.conf
```

Normal case - /usr/local/trip/sys/conf is a directory link  
Removing link from /usr/local/trip/sys/conf to:

```
/bigdisk/trip/v702/conf
Creating link /usr/local/trip/sys/conf to:
/bigdisk/trip/v702/conf
```

Calling /bigdisk/trip/v702/script/insttbs with parameters:  
/bigdisk/trip/v702 \_PREV\_TB\_USER\_ SunOS N

```
***** TRIPserver installation script *****
```

2014-02-17 12:54:40

Script called with:

```
Param. 1 TPATH = /bigdisk/trip/v702
Param. 2 TBUSER = _PREV_TB_USER_
Param. 3 OSNAME = SunOS
Param. 4 START_TND = N
```

Service pctdbs(n) already configured in file /etc/services :  
pctdbs 23457/tcp # TRIP Network Service ; 2014-02-17 10:27:17

Service "pctdbs" is already configured in file /etc/inetd.conf :  
pctdbs stream tcp nowait root /bigdisk/trip/v701/bin/tbserver  
tbserver

Copying file /etc/inetd.conf to /etc/cp.inetd.conf.1 for backup  
Commenting out service "pctdbs" in file /etc/inetd.conf

Copying file /etc/inetd.conf to /etc/cp.inetd.conf.2 for backup  
Adding service "pctdbs" last in file /etc/inetd.conf :  
pctdbs stream tcp nowait root /bigdisk/trip/v702/bin/tbserver  
tbserver

Restarting inetd

```
***** TRIPserver installation finished *****
```

TRIPserver installation successfully finished  
The TRIPnet Daemon (tripnetd) should NOT be started!

umask is reset to old value: 0002

The TRIP daemon (tripd) is not running. Trying to start it now!

The TRIP daemon (tripd) has been started.  
root 16161 1 0 12:54:45 ? 0:00  
/bigdisk/trip/v702/bin/tripd

\*\*\*\*\* The TRIPnet daemon (tripnetd) was not started.  
If you intend to use it you must start it manually  
to make TRIP available to the net/web users.

```
***** End of installing TRIP v7.0-2 on SunOS *****
```

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 for the TRIPserver (tbserver)

There is a TRIP utility, named stripping, that checks the different components that must be setup/running for tbserver to work.

(stripping = server-side TRIP ping)

This utility checks for correct installation. It verifies that:

- The services file contains the pctdbs service
- It is possible to connect to trip via the net
- tripd is installed and running
- inetd/xinetd is running

The program should be run it without arguments. A sample result from a run on a correctly set-up system is shown below (on Linux):

```
Checking services file... ok.
Checking connectability.. ok.
Checking tripd..... ok.
Checking xinetd..... ok.

TRIP is alive - reporting version 6.0-1
```

The following is a result from a run on a system where tripd is stopped (on Linux):

```
Checking services file... ok.
Checking connectability.. ok.
Checking tripd..... failed!

Daemon error: tripd is not running.
Checking xinetd..... ok.

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

The stripping utility is a simple and useful tool to verify that the TRIP installation is correctly set up.

If stripping reports that everything is OK and TRIPserver still does not work, please contact TRIP support for help.