Test Environment Toolkit

TETware Installation Guide
for the Windows NT and Windows 95 Operating Systems
Revision 1.5
TET3-IGN-1.5

Released: 19th November 1999

The Open Group

The information contained within this document is subject to change without notice.

This document is produced by UniSoft Ltd. at:

150 Minories
LONDON
EC3N 1LS
United Kingdom

CONTENTS

1. Introduction 1
   1.1 Preface 1
   1.2 Audience 1
   1.3 Conventions used in this guide 1
   1.4 Related documents 1
   1.5 Problem reporting 2

2. The TETware distribution 3
   2.1 Introduction 3
   2.2 The TET root directory 3
   2.3 Loading the distribution 3
   2.4 What to do next 6

3. Building TETware 7
   3.1 Introduction 7
   3.2 System requirements 7
   3.3 Before you build TETware 7
      3.3.1 Which version of TETware do you want to build? 7
   3.4 Preparing to build TETware 8
      3.4.1 Introduction 8
      3.4.2 Setting up the defined build environment 8
      3.4.3 The TETware configuration script — **`configure`** 9
      3.4.4 The **`tetconfig`** script 9
      3.4.5 The **`defines.mk`** file 10
         3.4.5.1 Introduction 10
         3.4.5.2 Support for Java 10
   3.5 Compiling the source 11
   3.6 Re-building TETware to use different options 12

4. Configuring your system to run Distributed TETware 13
   4.1 Introduction 13
   4.2 Services database entry 13
   4.3 The systems equivalence file 13
   4.4 Starting **`tccd`** 14
      4.4.1 Introduction 14
      4.4.2 Starting **`tccd`** on a Windows NT system 14

A. The TETware end-user licence 17

1. Introduction

1.1 Preface

There is one Installation Guide for each type of system on which TETware is supported. This document describes how to install both TETware-Lite and Distributed TETware on a Windows NT system, and how to install TETware-Lite on a Windows 95 system.

Throughout this document these operating systems are referred to collectively as **Win32 systems**. The individual system names are only used when it is necessary to distinguish between the two operating systems.

1.2 Audience

This document is intended to be read by software engineers and/or systems administrators who will install TETware on their computer systems. A knowledge of network administration is assumed when Distributed TETware is to be installed.

You may find it helpful to have a basic understanding of TETware architecture and its components before attempting to install TETware on your computer systems. This information is presented in the chapter entitled ‘‘TETware overview’’ in the TETware User Guide.

Test suite authors should refer to the TETware Programmers Guide for information about how to use the TETware Application Program Interfaces.

1.3 Conventions used in this guide

The following typographic conventions are used throughout this guide:

- `Courier font` is used for function and program names, literals and file names. Examples and computer-generated output are also presented in this font.
- The names of variables are presented in *italic font*. You should substitute the variable’s value when typing a command that contains a word in this font.
- **Bold font** is used for headings and for emphasis.

Long lines in some examples and computer-generated output have been folded at a \ character for formatting purposes. If you type such an example, you should type it in all on one line and omit the \ character.

1.4 Related documents

Refer to the following documents for additional information about TETware and its predecessors:

- *Test Environment Toolkit: TETware Programmers Guide*
- *Test Environment Toolkit: TETware User Guide*
- *Test Environment Toolkit: TETware Knowledge Base*

In addition, the TETware Release Notes contain important information about how to install and use TETware. You should read the release notes thoroughly before attempting to install and use each new release of TETware.

1.5 Problem reporting

If you have subscribed to TETware support and you encounter a problem while building or installing TETware, you can send a support request by electronic mail to the address given in the TETware Release Notes. Please follow the instructions contained in the release notes about how to submit such a request; in particular, please be sure to include all the information asked for by these instructions when submitting the request.

2. The TETware distribution

2.1 Introduction

This chapter describes what is in the TETware distribution and how to load it. You must load TETware on all the machines that you want to use to run local, remote or distributed tests.

2.2 The TET root directory

You must choose a directory on each machine in which to load the TETware distribution.

Note: Some of the file names in TETware are the same as those in other TET distributions. You should **not** load TETware in a directory that is occupied by an existing TET, DTET or ETET distribution, otherwise some files will be overwritten.

Each directory where TETware is loaded is known as the **TET root directory** and will be described here as *tet-root*. When you specify the path name of your TET root directory, replace *tet-root* with your equivalent path name.

2.3 Loading the distribution

You should load the TETware distribution on to every machine on which you want to run local, remote or distributed tests.

After you load the TETware distribution you will have a number of new files and directories on your machine. Some directories are present in both the source and the binary distributions, while others are present only in the source distribution.

The following list describes the directories which make up the distribution. Directories which are not present in the binary distribution are marked with a †. Directories which are empty in the source distribution but are populated when you build TETware are marked with a ‡. Directories for add-on products that may not be present in all distributions are marked with a §.

tet-root/`bin` Directory containing TETware executables.‡

tet-root/`demo` Test suite root directory for the example distributed test suite.[1]

tet-root/`doc/java` Directory that will contain `javadoc` output files when the Java API is installed.§

tet-root/`doc/ksh` Not used in TETware.[2]

tet-root/`doc/tet3` Directory subtree containing TETware documentation files.

tet-root/`doc/xpg3sh` Not used in TETware.[3]

[1] This directory is empty in both the source and the binary distribution. Instructions for running the distributed demonstration are presented in the TETware User Guide.

[2] The documentation below *tet-root*/doc/tet3 includes references to the Korn Shell API where appropriate. Thus, no separate documentation files appear in this directory.

[3] The documentation below *tet-root*/doc/tet3 includes references to the Shell API where appropriate. Thus, no separate documentation files appear in this directory.

tet-root`/inc/tet3` Directory containing public header files for use by test suites and report writers.

tet-root`/jdemo` Test suite root directory for the example Java test suite.§

tet-root`/lib/java` Directory containing library files for use when executing java test cases.‡§

tet-root`/lib/ksh` Directory containing library files for use when executing Korn Shell test cases.‡

tet-root`/lib/perl` Directory containing library files for use when executing Perl test cases.‡

tet-root`/lib/tet3` Directory containing library files for use when building TETware tools and C language test cases.‡

tet-root`/lib/xpg3sh` Directory containing library files for use when executing Shell test cases.‡

tet-root`/src/defines` Directory subtree containing example machine-specific makefile definition files.†

tet-root`/src/helpers` Directory containing shell scripts that are useful when preparing to build TETware.†

tet-root`/src/java` Source directory for the Java API.†§

tet-root`/src/ksh` Source directory for the Korn Shell API.†

tet-root`/src/perl` Source directory for the Perl API.†

tet-root`/src/scripts` Source directory for shell scripts.†

tet-root`/src/tet3/apilib`

Directory containing source files for the C API.†

tet-root`/src/tet3/apishlib`

Directory used when building the shared version of the C API library.†

tet-root`/src/tet3/apithr`

Directory containing source files for the Thread-safe C API.†

tet-root`/src/tet3/apithrshlib`

Directory used when building the shared version of the Thread-safe C API library.†

tet-root`/src/tet3/bin` Directory containing scripts that are used when building TETware.†

tet-root`/src/tet3/demo/master`

Directory containing source files for the master part of a simple demonstration distributed test suite.

tet-root`/src/tet3/demo/slave`

Directory containing source files for the slave part of a simple demonstration distributed test suite.

tet-root`/src/tet3/dtet2lib`

Directory containing source files for low-level library functions.†

tet-root`/src/tet3/inc` Directory containing private header files for the C sources.†

tet-root`/src/tet3/inetlib`

Directory containing source files for INET transport-specific library functions.†

tet-root`/src/tet3/japilib`

Directory containing source files for the Java API.†§

tet-root`/src/tet3/jtools`

Directory containing source files for the tools that are used when processing Java test cases.†§

tet-root`/src/tet3/llib` Directory containing lint library source files.

tet-root`/src/tet3/servlib`

Directory containing source files for the generic client/server code and server access library functions.†

tet-root`/src/tet3/syncd` Directory containing source files for the Synchronisation daemon `tetsyncd`.†

tet-root`/src/tet3/tcc` Directory containing source files for the Test Case Controller `tcc`.†

tet-root`/src/tet3/tccd` Directory containing source files for the Test Case Controller daemon `tccd`.†

tet-root`/src/tet3/tcclib`

Directory containing source files for `tcc` action functions.†

tet-root`/src/tet3/tcm` Directory containing source files for the C and C++ Test Case Manager components.†

tet-root`/src/tet3/tcmshlib`

Directory used when building versions of the C and C++ Test Case Managers that are used with the shared C API library.†

tet-root`/src/tet3/tcmthr`

Directory used when building the Thread-safe C and C++ Test Case Managers.†

tet-root`/src/tet3/tcmthrshlib`

Directory used when building versions of the C and C++ Test Case Managers that are used with the shared Thread-safe C API library.†

tet-root`/src/tet3/xresd` Directory containing source files for the Execution Results daemon `tetxresd`.†

tet-root`/src/tet3/xtilib`

Directory containing source files for XTI transport-specific library functions.†

tet-root`/src/xpg3sh` Source directory for the Shell API.†

2.4 What to do next

If you have loaded a TETware source distribution, you should build and configure TETware on your system by working through the instructions presented in the rest of this guide.

If you have loaded Distributed TETware from a binary distribution, you should configure TETware on your system. Details of how to do this are presented in the chapter entitled ‘‘Configuring your system to run Distributed TETware’’ later in this guide.

If you have loaded TETware-Lite from a binary distribution, no further configuration is required and it is now ready to use.

3. Building TETware

3.1 Introduction

This chapter describes how to build TETware from a source distribution. If you are installing TETware from a binary distribution you should skip to the next chapter.

You should build TETware on every machine on which you want to run local, remote or distributed tests.

3.2 System requirements

Prior to building TETware on each machine, you should check that the following requirements are met as TETware will not build without them. A defined build environment is required in order to build TETware on a Win32 system. This consists of the following software:

- Microsoft Visual C++ version 4.0 or later.
- The MKS Toolkit[9] for Windows NT systems.

The MKS Toolkit is also required in order to use the Shell (`xpg3sh`) and Korn Shell (`ksh`) APIs.

A Perl implementation for Windows NT is required in order to build and use the Perl API. A suitable version may be obtained by FTP from `ntperl.hip.com`.[10]

Distributed TETware uses the Windows Socket (Winsock) network API for interprocess communication.

The Java Development Kit (JDK) version 1.1 is required in order to build and use the Java API.

3.3 Before you build TETware

3.3.1 Which version of TETware do you want to build?

You must first decide whether you want to build TETware-Lite or Distributed TETware. A description of the differences between these two options is presented in the chapter entitled ‘‘TETware overview’’ in the TETware User Guide.

Please note that although it is possible to build Distributed TETware on a Windows 95 system, Distributed TETware can only be used on a Windows NT system. So if you intend to use TETware on a Windows 95 system, you should select the configuration option which builds TETware-Lite.

[9] MKS, MKS Make and MKS Toolkit are trademarks of Mortice Kern Systems Inc.
The version used during development is MKS Toolkit Release 6.1.
The Windows NT version of the MKS Toolkit may also be used on Windows 95 systems.

[10] This version of Perl is also suitable for use on Windows 95 systems.

3.4 Preparing to build TETware

3.4.1 Introduction

Once you have made the decisions described above, you can prepare to build TETware with the makefiles supplied. The following subsections describe what you must do on each machine in order to configure TETware in accordance with the decisions that you have made.

3.4.2 Setting up the defined build environment

A build environment has been defined for use when building TETware on a Win32 system. The purpose of defining this environment is to enable TETware to be built on a Win32 system using a method which resembles the one used on UNIX systems as closely as possible. Another benefit of this approach is that the changes that must be made to enable TETware source code and makefiles to be usable on a Win32 system have been kept to a minimum.

Before you can build TETware on a Win32 system, you must first prepare the defined build environment. When working with TETware on a Win32 system you should always use the MKS Korn Shell (`sh`) as your command interpreter.

When defining an environment variable it is best to include a definition for the variable in your profile file `$HOME/profile.ksh` so that it is available each time that you start a shell.

You should perform the following actions to set up the defined build environment:

1. Install Microsoft Visual C++ and the MKS Toolkit if you have not already done so. When installing the MKS Toolkit it is best if you install it in the root directory of the default disk drive; i.e., `c:/`. Each instruction in this document that is specific to Win32 systems assumes that you have installed the Toolkit in this location.

2. Ensure that you always use the MKS Korn Shell (`sh`) as the command interpreter when performing any TETware operations.

3. Ensure that your `PATH` environment variable includes the MKS `bin` directory which is usually called `c:/mksnt`.

4. MKS `make` uses a configuration file to adapt itself to the compiler which is to be used[13] (defaults to `c:/etc/startup.mk`).
   You should **either**:

   i. Copy the Microsoft Visual C++ `make` startup file `c:/etc/msc.mk` to `c:/etc/startup.mk`.

   or:

   ii. Set the `MAKESTARTUP` environment variable to `c:/etc/msc.mk` and export it.

[13] Refer to the `make(1)` manual page near the end of the MKS Make User Guide for details.

5. MKS `cc` uses a configuration file to adapt itself to the compiler which is to be used.[14] A customised compiler configuration file for use when building TETware is supplied with the distribution. The name of this file is *tet-root*`/src/tet3/compiler.ccg`. You should set the `CCG` environment variable to the name of this file and export it.

6. Ensure that the directory `c:/tmp` exists and is accessible to all users.

3.4.3 The TETware configuration script — `configure`

The TETware distribution includes a configuration script which can be used to perform the tasks described in the next few sections. The name of this script is `configure` and it resides in the *tet-root* directory. This script attempts to guess the type of operating system on which it is running, invokes the `tetconfig` script and installs a suitable `defines.mk` file. (The `tetconfig` script and the `defines.mk` file are described in the next two sections.)

You should perform the following operations to run the `configure` script on each system:

1. To start, if you are not already there, change directory to *tet-root* thus:

   ```
   cd tet-root
   ```

2. Type

   ```
   sh configure -t transport
   ```

   where *transport* is one of `inet` to build Distributed TETware or `lite` to build TETware-Lite.

If you are building Distributed TETware and `configure` is successful in configuring the TETware source tree, you can skip the next few sections and resume at the section entitled ‘‘Compiling the source’’ later in this chapter. Otherwise, `configure` will outline the things that you must do before TETware can be built. Details of these things are presented in the sections that follow.

3.4.4 The `tetconfig` script

If you did not use the `configure` script described in the previous section, you must run the configuration script in the file *tet-root*`/src/tetconfig`. This script asks you which network interface you want to use when building Distributed TETware, or whether you want to build TETware-Lite. You should reply with `inet` to build Distributed TETware or `lite` to build TETware-Lite. You should select `lite` if you intend to run TETware on a Windows 95 system. You should not select the `xti` option since it is not supported on Win32 systems.

You should perform the following operations to run `tetconfig` on each system:

1. To start, if you are not already there, change directory to *tet-root*`/src` thus:

   ```
   cd tet-root/src
   ```

[14] Refer to the `cc(1)` manual page near the end of the MKS Make User Guide for details.

2. Type

   ```
   sh tetconfig
   ```

3. You must then select which version of TETware that you want to build. You should respond with `inet` to build Distributed TETware using Winsock, `lite` to build TETware-Lite, or `q` to quit without configuring TETware.

Each source directory contains makefile **include** files which contain transport-specific makefile fragments for TETware-Lite and each supported network interface. The TETware configuration script arranges for the appropriate **include** file to be picked up by each directory's makefile.

3.4.5 The `defines.mk` file

3.4.5.1 Introduction

There is a makefile definition file that you must supply on each system before TETware can be built; the name of this file is *tet-root*`/src/defines.mk`. This file contains system-specific definitions and is **include**d in the makefiles that are used to build TETware.

Several example makefile definition files suitable for use with various systems are included with the distribution and may be found in the directory *tet-root*`/src/defines`.

If you are building TETware on a Win32 system using the defined build environment[15] you should copy the supplied makefile definition file *tet-root*`/src/defines/mks+msc.mk` to *tet-root*`/src/defines.mk`. If you have decided to build TETware-Lite, edit *tet-root*`/src/defines.mk` and remove `wsock32.lib` (the Winsock library) from the `SYSLIBS` definition.

The Win32 version of the TETware source code has been designed to compile in the defined build environment and the use of a different build environment is outside the scope of the TETware software support service. If you decide to attempt to build TETware using a different build environment for some reason you will probably need to create your own `defines.mk` file. You should refer to the section entitled ‘‘The `defines.mk` file’’ in the TETware Installation Guide for UNIX Operating Systems for information on how to do this.

3.4.5.2 Support for Java

When the Java API source files are included in the distribution, it is possible to build the Java API on platforms on which the JDK is installed.

When you configure your system to build the Java API it is helpful to understand the following points:

1. The TETware Java API consists of the following parts:

   - Java class files.
     These are built from Java source files and are put in an archive file.

   - The API libraries.
     These libraries are built from most of the source files that are used to build the

[15] That is: Microsoft Visual C++ and the MKS Toolkit.

C API, together with some C files which are specific to the Java API. These files are compiled into shared libraries which are accessed by the Java runtime using the Java Native Interface (JNI).

2. Since on Win32 platforms there is no ‘‘standard’’ place to install the JDK, the locations of the Java include files cannot be specified in the example `defines.mk` file that is included with the TETware distribution. Instead, you will need to specify these locations on each system on which you build the API.

If the Java compiler `javac` can be executed and the directory *tet-root*`/src/java` is present, the API class files will be built.

The variable `JAVA_CDEFS` is used to determine whether or not the API libraries and support tools should be built. If you do not want to build the Java API libraries and support tools, you should make the following assignment:

```
JAVA_CDEFS = JAVA_NOT_SUPPORTED
```

Otherwise, to build the API libraries and support tools you should use `JAVA_CDEFS` to specify set of compiler definitions to be used **as well as** `TET_CDEFS` or `DTET_CDEFS` when compiling the API support library. This should include the following values:

- A `−I` option that specifies the location of the generic Java include files.
- A `−I` option that specifies the location of the platform-specific Java include files.

For example, if the JDK has been installed in `c:/jdk1.1`, you would make the following assignment:

```
JAVA_CDEFS = -Ic:/jdk1.1/include -Ic:/jdk1.1/include/win32
```

It is not necessary to change the other `make` variables under the heading ‘‘support for Java’’ in the Win32 `defines.mk` file.

3.5 Compiling the source

When all the `make` definitions are correct, you may build TETware.

Note: Ensure that you have run the *tet-root*`/src/tetconfig` command and installed a suitable `defines.mk` file before you start on this section.[18]

You should perform the following operations on each system:

1. To start, if you are not already there, change directory to *tet-root*`/src` thus:

```
cd tet-root/src
```

[18] Either by running the `configure` script or by performing these operations by hand.

2. Type

   ```
   make install
   ```

   to build the tools and APIs and install them in their destination directories under *tet-root*.

3.6 Re-building TETware to use different options

If, having once built TETware, you decide that you want to switch between TETware-Lite and Distributed TETware, you should perform the following operations:

1. Ensure that `tccdstart` is not running.
2. Ensure that there is no `tccd` currently running.
3. Change directory to *tet-root*`/src`, thus:

   ```
   cd tet-root/src
   ```

4. Type

   ```
   make clobber
   ```

   to remove the API library files and all the TETware object files and executables.[19]
5. Refer back to the section entitled ‘‘Before you build TETware’’ earlier in this chapter, and follow all the instructions from that point up to and including the section entitled ‘‘Compiling the source’’ above.

[19] Be sure to perform this action **before** you use the `tetconfig` script to configure TETware for your new option.

4. Configuring your system to run Distributed TETware

4.1 Introduction

If you have installed Distributed TETware, you must now configure each system to run TETware. If you have installed TETware-Lite, no configuration is necessary and so you do not need to perform the operations described in this chapter.

You may need to have administrative privilege in order to perform some of the operations described in this chapter.

Since Distributed TETware can only be run on a Windows NT system, the information presented in this chapter is not applicable to a Windows 95 system.

4.2 Services database entry

You must add an entry to the **services** database[20] on each system where Distributed TETware is installed.

The Test Case Controller daemon bootstrap program `tccdstart` listens for requests on the well-known Internet TCP port number specified for the **tcc** service in the **services** database. This port number must be the same on all systems that are to participate in a set of remote or distributed tests. The **tcc** port number should be that of a non-privileged port (i.e., 1024 or greater).

For example, to define the well-known port as TCP port 1234, you might add the following line to the **services** database on each TETware system:

```
tcc     1234/tcp
```

4.3 The systems equivalence file

You must create a file called `systems.equiv` which specifies the names of systems from which `tccd` may accept connection requests. This file resides in the directory specified by the `HOME` environment variable which is in effect when `tccd` is started by `tccdstart`.

You should create an entry in this file for each system that is to be permitted to send requests to `tccd` on this system. `tccd` will only process requests from another system if an entry for that system appears in the `systems.equiv` file.

An example `systems.equiv` file is included in the TETware distribution. You may copy this file to the `tccd` user's home directory and edit it as required.

Information about the format of the `systems.equiv` file is presented in a manual page at the back of the TETware User Guide.

[20] On Windows NT version 3.51, this database usually resides in the file `c:/winnt35/system32/drivers/etc/services`.

On Windows NT version 4.0, this database usually resides in the file `c:/winnt/system32/drivers/etc/services`.

4.4 Starting `tccd`

4.4.1 Introduction

You should make arrangements for `tccd` to be started on each machine where Distributed TETware is installed.

Note: You must start `tccd` on the local system as well as on remote systems.

Further information about `tccd` is presented in a manual page at the back of the TETware User Guide.

You should **not** arrange to start `tccd` on a machine connected to an external network without first considering the security implications of doing so. This issue is discussed further in the section entitled ‘‘Network security considerations for Distributed TETware’’ in the TETware User Guide.

4.4.2 Starting `tccd` on a Windows NT system

You should ensure that the directory `c:/tmp` exists, creating it if necessary.

On a Windows NT system, `tccd` is started on demand by a bootstrap program called `tccdstart` which operates in a way similar to `inetd` on a UNIX system. To arrange for `tccd` to start on demand, you should open a new Korn Shell window and type

```
tccdstart
```

at the command prompt. If all is well you will see a startup message printed on the standard output.

Further information about `tccdstart` is contained in a manual page at the back of the TETware User Guide.

APPENDICES

A. The TETware end-user licence

++++++++++++++++++++++++++++ TET END USER LICENCE ++++++++++++++++++++++++++++

BY OPENING THE PACKAGE, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, DO NOT INSTALL THE PRODUCT AND RETURN IT TO THE PLACE OF PURCHASE FOR A FULL REFUND.

TETWARE RELEASE 3.4 END USER LICENCE
REDISTRIBUTION NOT PERMITTED

This Agreement has two parts, applicable to the distributions as follows:

A. Free binary evaluation copies – valid for 90 days, full functionality – no warranty.

B. Free binary restricted versions – no warranty, limited functionality.

C. Licenced versions – full functionality, warranty fitness as described in documentation, includes source, binary and annual support.

PART I (A & B above) – TERMS APPLICABLE WHEN LICENCE FEES NOT (YET) PAID (LIMITED TO EVALUATION, EDUCATIONAL AND NON-PROFIT USE).

GRANT.

X/Open grants you a non-exclusive licence to use the Software free of charge if

a. you are a student, faculty member or staff member of an educational institution (K-12, junior college, college or library) or an employee of an organisation which meets X/Open’s criteria for a charitable non-profit organisation; or

b. your use of the Software is for the purpose of evaluating whether to purchase an ongoing licence to the Software.

The evaluation period for use by or on behalf of a commercial entity is limited to 90 days; evaluation use by others is not subject to this 90 day limit. Government agencies (other than public libraries) are not considered educational or charitable non-profit organisations for purposes of this Agreement. If you are using the Software free of charge, you are not entitled to hard-copy documentation, support or telephone assistance. If you fit within the description above, you may use the Software for any purpose and without fee.

DISCLAIMER OF WARRANTY.

Free of charge Software is provided on an ‘‘AS IS’’ basis, without warranty of any kind.

X/OPEN DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL X/OPEN BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

PART II (C above) – TERMS APPLICABLE WHEN LICENCE FEES PAID.

GRANT.

Subject to payment of applicable licence fees, X/Open grants to you a non-exclusive licence to use the Software and accompanying documentation (‘‘Documentation’’) as described below.

LIMITED WARRANTY.

X/Open warrants that for a period of ninety (90) days from the date of acquisition, the Software, if operated as directed, will substantially achieve the functionality described in the Documentation. X/Open does not warrant, however, that your use of the Software will be uninterrupted or that the operation of the Software will be error-free or secure.

SCOPE OF GRANT.

Permission to use for any purpose is hereby granted. Modification of the source is permitted. Redistribution of the source code is not permitted without express written permission of X/Open. Distribution of sources containing adaptations is expressly prohibited.

Redistribution of binaries or binary products containing TETware code is permitted subject to the following conditions:

— this copyright notice is included unchanged with any binary distribution;

— the company distributing binary versions notifies X/Open;

— the company distributing binary versions holds an annual TET support agreement in effect with X/Open for the period the product is being sold, or a one off binary distribution fee equal to four years annual support is paid.

Modifications sent to the authors are humbly accepted and it is their prerogative to make the modifications official.

Portions of this work contain code and documentation derived from other versions of the Test Environment Toolkit, which contain the following copyright notices:

The unmodified source code of those works is freely available from `ftp.xopen.org`. The modified code contained in TETware restricts the usage of that code as per this licence.

++