Test Environment Toolkit

Release Notes for TETware Release 3.3
TET3-RN-3.3

Released: 18th September 1998

The Open Group

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

Copyright © 1996,1997 X/Open Company Limited

All rights reserved. No part of this source code or documentation may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as stated in the end-user licence agreement, without the prior permission of the copyright owners. A copy of the end-user licence agreement is contained in the file `Licence` which accompanies the TETware distribution.

Motif, OSF/1, UNIX® and the 'X' device are registered trademarks and IT DialTone and The Open Group are trademarks of The Open Group in the US and other countries.

X/Open is a trademark of X/Open Company Limited in the UK and other countries.

Win32™, Windows NT™ and Windows 95™ are registered trademarks of Microsoft Corporation.

This document is produced by UniSoft Ltd. at:

150 Minories
LONDON
EC3N 1LS
United Kingdom

Tel: +44 171 264 2120
Email: `tet@root.co.uk`

++++++++++++++++++++++++++ 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 3.3 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.

Copyright © 1996,1997 X/Open Company Ltd.
Copyright © 1998 The Open Group

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:

Copyright © 1990,1992 Open Software Foundation
Copyright © 1990,1992 Unix International
Copyright © 1990,1992 X/Open Company Ltd.
Copyright © 1991 Hewlett-Packard Co.
Copyright © 1993 Information-Technology Promotion Agency, Japan
Copyright © 1993 SunSoft, Inc.
Copyright © 1993 UNIX System Laboratories, Inc., a subsidiary of Novell, Inc.
Copyright © 1994,1995 UniSoft Ltd.

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.

++

1. Introduction

1.1 Preface

These release notes accompany TETware Release 3.3.

TETware includes all of the functionality of the Test Environment Toolkit Release 1.10 (TET), the Distributed Test Environment Toolkit Version 2 Release 2.3 (dTET2) and the Extended Test Environment Toolkit Release 1.10.3 (ETET), together with a number of new features.

There are two principle versions of TETware. One version is known as Distributed TETware. This version uses a client-server architecture and provides support for processing local, remote and distributed test cases. The other version is known as TETware-Lite. This version does not use a client-server architecture or require a network transport. It is more simple to set up than is Distributed TETware but does not provide support for the processing of remote or distributed test cases.

Distributed TETware is implemented on UNIX operating systems and also on the Windows NT operating system. TETware-Lite is implemented on UNIX operating systems and also on the Windows NT and Windows 95 operating systems.

Throughout this document, the Windows NT and Windows 95 operating systems are referred to collectively as **Win32 systems**. The individual system names are only used when it is necessary to distinguish between them.

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 system administration is assumed when TETware installation and configuration instructions are presented. In addition, a knowledge of network administration is assumed when TETware is to be built to use network transports.

1.3 Conventions used in this document

The following typographic conventions are used throughout this document:

- `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.

1.4 Related documents

Refer to the following documents for additional information about TETware:

- *Test Environment Toolkit: TETware Installation Guide*
  There is one version of this document for each operating system family on which TETware is implemented.
- *Test Environment Toolkit: TETware User Guide*

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

Source and postscript versions of these documents are included in the TETware distribution.

1.5 Problem reporting

If you have subscribed to TETware support and you encounter problems when building or executing TETware you should take a copy of the error reporting template contained in the file *tet-root*`/doc/tet3/err.template` in the distribution, fill in the details of the problem, and send it by electronic mail to:

`tet_support@xopen.co.uk`

Alternately you may use a web browser to complete and submit the HTML form included in the file *tet-root*`/doc/tet3/errtemp.html` in the distribution.

You should include sufficient information in your report to enable someone who is unfamiliar with your system to be of assistance in solving the problem.

Users are reminded that the user-contributed software which accompanies the TETware distribution is not covered by TETware support services.

2. New features in this release

The following features appear for the first time in this release of TETware:

- Support is provided to enable shared versions of the C API libraries to be built. The shared library schemes that are supported on UNIX systems are described in the chapter entitled ‘‘Support for shared API libraries’’ in the TETware Installation Guide for UNIX Operating Systems. On Win32 systems support is provided to build DLL versions of the API libraries.

  Information regarding the use of shared API libraries is presented in the chapter entitled ‘‘Using shared API libraries’’ in the TETware Programmers Guide.

- A document has been added to the TETware document set; its title is TETware Knowledge Base. It contains information extracted from replies to requests that have been handled by the TETware support service. An HTML version of this document is included in the distribution in addition to the usual PostScript version.

- A shell script exec tool has been added to the distribution which enables user interaction with test cases when Distributed TETware is used. The name of this tool is `tet_start` and it is described in the chapter entitled ‘‘Interacting with test cases in Distributed TETware’’ in the TETware User Guide.

- It is now possible to specify `tccd`’s well-known port number in the `systems` file when Distributed TETware is built to use the socket network interface. Thus it is possible to have `tccd` listening on different port numbers on different systems, or to have several instances of `tccd` running with different options on a particular system.

- Support for multiple network interfaces has been added when Distributed TETware is built to use the socket interface. This support is provided through the `TET_LOCALHOST` distributed configuration variable.

Note: The circumstances under which you might need to specify `TET_LOCALHOST` are rare when Distributed TETware is built to use the socket interface.[1] Do not specify this variable unless you really need it.

- A keyword `%include` has been added to the scenario language. This keyword can be used in a scenario file in much the same way as `#include` is used in a C program.[2]
- A configuration variable expansion mechanism has been added to `tcc`. This feature is described in the section entitled ‘‘Configuration variable expansion’’ in the TETware Programmers Guide.
- It is now possible to specify a number range with the `:remote:` and `:distributed:` scenario directives.
- The maximum number of test cases that `tcc` can process in parallel has been increased from 26 to 676.
- New makefile variables have been introduced to provide a finer degree of control over the specification of compiler options when the Thread-safe APIs are built. The names of these variables are `TET_THR_CDEFS` and `DTET_THR_CDEFS`, and they are described in the section entitled ‘‘Support for Threads’’ in the TETware Installation Guide for UNIX Operating Systems.
- A command-line option has been added to `tcc`. When `tcc` is invoked with the `–V` option, it prints out version information and exits.
- A shell script called `configure` has been added to the distribution in the *tet-root* directory. On platforms for which a suitable `defines.mk` file is provided in the distribution this script automates the tasks that must be performed before TETware can be built.
- A shell script called `sigtool` has been added to the distribution in the *tet-root*`/src/helpers` directory. This script is useful when preparing to build TETware on a platform for which no `defines.mk` file is provided in the distribution. The function of this script is to generate values for makefile variables that must be specified for use when the Shell APIs are built.
- The header files `inc/tet3/tet_api.h` and `inc/tet3/tet_jrnl.h` are now protected against multiple inclusion.
- The Korn Shell API has been updated to work with `ksh93`.
- A number of problems reported after the previous TETware release have been fixed. All the source code patches issued since the last TETware release have been incorporated into this release.
- The documentation has been revised and updated.

---

1. By contrast, `TET_LOCALHOST` must always be specified when Distributed TETware is built to use XTI.
2. Note that the `%include` keyword provides different functionality from that provided by the `:include:` scenario directive.

3. Status of this release

This release of TETware is a general release for production use.

4. Problems fixed since the last release

The following problems have been fixed since the last TETware release:

- When a test purpose failed to register a result in Distributed TETware, the result generated by the API was `UNRESOLVED` instead of `NORESULT`.
- When the underlying call to `execve()` failed in a call to `tet_spawn()` on a UNIX system, the error condition was not returned to the process that called `tet_spawn()`.
- When remote (non-distributed) test cases were processed on more than one system, the test cases on the second and subsequent systems did not print TCM Start, IC Start and IC End messages to the journal, and a call to `tet_result()` failed with an `ER_DONE` error.
- When a ^*scenario-name* appeared within the scope of a `:distributed:` scenario directive, test cases in the scenario referenced by *scenario-name* were unable to synchronise with each other.
- Under certain circumstances it was possible for `tcc` to terminate with a memory fault when removing the temporary directory after processing a test case in execute mode.
- When the `TET_RUN` environment variable was specified, `tcc` attempted to create a file with a garbage name on some platforms.
- When `tcc` processed a test case in execute mode with `TET_EXEC_IN_PLACE` set to `False`, the test case was copied to a location immediately below the temporary directory and executed from there rather than to a location below the temporary directory which included the path name of the test case relative to the test suite root directory. Thus it was not possible for a test case to use the `tet_pname` variable in the C API to determine its full path name relative to the test suite root directory.
- The thread cleanup code in the Thread-safe API assumed that a thread ID would not be re-used by the implementation immediately after the corresponding thread had been joined. This caused problems on threads implementations where this assumption is not true.
- If none of the test cases within the scope of a `:timed_loop:` directive can be built, the loop is effectively empty during execute mode. Under these circumstances, `tcc –be` would thrash when in execute mode (possibly generating huge volumes of journal output).
- There was a name space clash when building the Thread-safe APIs on some platforms which support both UI threads and POSIX threads.
- If a locking failure prevented `tcc –be` from building a test case, `tcc` still attempted to execute the test case that could not be built.
- There were a few references in the code to the non-standard interfaces `sys_errlist` and `sys_nerr`. The functions that referred to these symbols have been re-written to make calls to `strerror()` instead.
- Descriptions of compiler options used when building Distributed TETware to use XTI were missing from the TETware Installation Guide for UNIX Operating Systems.

- In Distributed TETware, if a configuration variable exchange between the local system and a remote system failed because the related configuration file was missing on the remote system, the error message printed by `tcc` was not particularly useful. The error message now states that the remote configuration file could not be opened.
- A feature of the Linux implementation of `getopt()` caused `tcc` to terminate with a memory fault when `tcc` was invoked in rerun or resume mode. The code that calls `getopt()` has been changed to work around this feature.
- The `–I` option was missing in `tcc`'s usage message.

5. Known problems in this release

The following problem has been reported too late for the fix to be included in this release:

- When `make clobber` is executed in the *tet-root*`/src` directory, any compatibility links that have been installed by a previous `make compat` are not removed.

6. Building and installing TETware

6.1 Building and installation instructions

For information on how to build and install TETware, please following the instructions in the version of the TETware Installation Guide which is appropriate for your system.

Note: There have been significant changes to the makefile scheme in order to support some of the new features introduced in this release. All the `defines.mk` files that are supplied with the distribution have been updated to support the new scheme. However, if you use your own `defines.mk` file when building TETware, you will need to add some extra variable assignments to the file before you can use it with the new scheme.

You should add assignments for all the variables in the following list. If you decide not to make use of any of the new features in this release that use these variables, the values shown here will enable your `defines.mk` file to continue to work with the new makefile scheme.[3]

```
TET_THR_CDEFS = $(DTET_CDEFS)
DTET_THR_CDEFS = $(DTET_CDEFS)
SHLIB_COPTS = SHLIB_NOT_SUPPORTED
SHLIB_CC =
SHLIB_BUILD =
SHLIB_BUILD_END =
THRSHLIB_BUILD_END =
SH =
SO =
```

3. The values shown here are correct for UNIX systems. The file *tet-root*`/src/defines/msc+mks.mk` contains values that should be used with the defined build environment on a Win32 system.

These variables are described in the section entitled ‘‘The `defines.mk` file’’ in the TETware Installation Guide for UNIX Operating Systems.

6.2 Installed platforms

6.2.1 UNIX systems

Both TETware-Lite and Distributed TETware versions have been installed and tested on the following platforms:

- AIX Release 4.3.1 using `c89`
- HP-UX Release 10.01 and Release 10.10
- Linux version 2.0.30 (Slackware) using `gcc` version 2.7.2
- Linux version 2.0.34 (RedHat 5.1) using `gcc` version 2.7.2.3 and `glibc2`
- NCR UNIX System V Release 4.0 with the MetaWare High C compiler
- OSF/1 using `c89`
- Solaris Release 2.6 using the Sun Workshop Compiler and using `gcc` version 2.8.0
- UNIX System V Release 4
- UnixWare Release 2.1.2

6.2.2 Win32 systems

TETware-Lite and Distributed TETware have been installed and tested on Intel PCs running Windows NT release 4.0.

TETware-Lite has been installed and tested on Intel PCs running Windows 95.

6.3 API status

The Thread-safe APIs have been exercised using POSIX threads and UI threads on Solaris, and using UI threads on UnixWare.

The following behaviour has been observed when using the Thread-safe API on UnixWare:

i. A call to `tet_printf()` in a non-main thread causes the process to receive a `SIGSEGV` signal.

ii. Cleanup of left-over threads sometimes causes a fatal error.

It is believed that this behaviour may be due to a problem with the threads implementation on this platform.

On Win32 systems the thread-safe APIs must be used with the multi-threaded DLL version of the C runtime support library. Use with the multi-threaded static version of the C runtime support library is not supported.

The C++ APIs have been exercised on Linux, Solaris, UnixWare and Win32 systems.

6.4 Transport-specific status

6.4.1 Socket network interface

Versions of Distributed TETware using the socket network interface have been tested on all the platforms listed previously.

6.4.2 XTI network interface

Versions of Distributed TETware using the XTI network interface have been tested on AIX 4.3, Solaris 2.6 and UnixWare 2.1.1 using TCP as the underlying transport provider.

There is a known problem with some SVR4 XTI implementations in which the `t_sync()` function does not work correctly. An XTI implementation may hold transport endpoint data either in kernel or in user address space. The `t_sync()` function is included in the XTI specification for the benefit of implementations which do not automatically detect when the transport endpoint data held in user space is lost. Examples of when such loss might occur are when the file descriptor underlying a transport endpoint is duplicated using `fcntl()` or when the process address space is overlaid by one of the `exec()` system calls.

Since the XTI version of TETware may perform both of these operations, it will not function when used with an XTI implementation which does not automatically detect the loss of data held in user address space and in which `t_sync()` does not perform the advertised function. A common symptom of this problem is when one of the servers `tetsyncd` and `tetxresd` fails with a `TBADQLEN` error associated with a `t_listen()` call soon after being started by `tcc`.

The XTI version of Distributed TETware has not been tested using an OSI connection orientated transport provider.

7. TETware for TET and ETET users

Some hints and tips for users experienced with previous TET implementations are presented in an appendix to the TETware User Guide.