

Test Environment Toolkit

Release Notes for TETware Release 3.2 TET3-RN-3.2

Released: 31st July 1997

X/Open Company Limited

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 this distribution.

X/Open and the 'X' symbol are trademarks of X/Open Company Limited in the UK and other countries.

UNIX[®] is a registered trademark in the U.S. and other countries, licenced exclusively through X/Open Company Ltd.

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

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 3.2 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.

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 3.2.

TETware includes all of the functionality of 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 and its predecessors:

- *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*

Source and postscript versions of these documents are included in the source 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 is included in 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 TETware-Lite to run on Windows 95 systems.
- The thread-safe APIs have been extended so as to enable them to be used in conjunction with the multi-threaded DLL version of the C runtime support library on Win32 systems.
- A new function in the thread-safe APIs — `tet_beginthreadex()` — enables a new thread to be created on a Win32 system.
- A dynamic interface has been introduced between the C TCM and the test case as an alternative to the static interface provided by the `tet_testlist[]` array. Use of the dynamic interface enables a test case to be written where the list of invocable components and test purposes is not defined at compile time by the test suite author but instead is defined at runtime by the test case itself.
- A `tcc` command-line option has been added which causes the journal lines which mark the start and end of test case start, execute and clean operations to be printed on the standard error stream as well as in the journal.
- In the `perl` API, a call to `tet_reason` now returns the deletion reason string instead of printing it on the standard output. This change has been made because the previous behaviour (which was inherited from the Shell API) is considered not to be useful in the environment provided by the `perl` language.
- A clause has been added to the TET End-user licence which covers the re-distribution of TETware binaries and binary products containing TETware code.
- 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.

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:

- A problem has been fixed which prevented `tcc` from processing a remote test suite (or test suite part) residing on a read-only file system even when `TET_REMnnn_TET_RUN` was specified.
- In Distributed TETware, `tcc` now checks the environment for badly formatted strings and excludes them when passing the environment to `tccd` on the local system. This problem can occur when `tcc` is invoked from `ssh`, which sometimes puts garbage in the environment.
- A problem has been fixed which prevented `tcc` from removing the temporary execution directory on a Win32 system if the directory contained read-only files.
- When `TET_EXEC_IN_PLACE` is false, the copy to the temporary execution directory is now performed from the alternate execution directory if one is specified. Previously the copy was always performed from the test case source directory irrespective of whether or not an alternate execution directory had been specified.
- Leading white space is now permitted on lines in a scenario include file, in order to provide backward compatibility with previous TET implementations.
- `tcc` now prints a System Information line to the journal after the TCC Start line, in order to provide backward compatibility with previous ETET releases.
- When remote or distributed test cases appear within the scope of a `parallel` directive, they are now always processed in parallel. Previously there were some circumstances which could cause such test cases to be processed sequentially.
- A problem has been fixed in the `perl` API which caused only the first word of a test purpose's deletion reason to be printed in the journal after the test purpose had been deleted.
- A problem has been fixed in the `perl` API which caused the deletion or undeletion of a test purpose to affect the deletion status of other similarly-named test purposes under certain circumstances. For example, it was possible to delete or undelete `tp1` and find that `tp10`, `tp11`, `tp100` etc. had been undeleted as well.
- A problem has been fixed which caused `tccdstart -p port-number` to terminate with a memory fault on a Windows NT system.
- A problem has been fixed in `tcc`'s signal blocking emulation code on Win32 systems which caused the arrival of a second or subsequent keyboard signal to perform the wrong action.
- The context number in a child process started by a call to `tet_spawn()` is now different from that of its parent on Win32 systems.

- A problem has been fixed which caused a child process started by `tet_spawn()` on a Win32 system to terminate with an error message reporting an incorrect argument count.

5. Known problems in this release

There are no known problems in this release.

6. Building and installing TETware

6.1 Building and installation instructions

For instructions 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.

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 4.1.1 using `c89`
- AIX Release 4.2 using `gcc` version 2.7.2.2
- HP-UX Release 10.01
- Linux version 2.0.27 (RedHat 4.1) using `gcc` version 2.7.2.1
- OSF/1 using `c89`
- Solaris Releases 2.4 and 2.5
- UnixWare Releases 2.03 and 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 release 4.0.

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, 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 System V Release 4 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.

