

# TETware

## **GUI User Guide**

for GUI Version 1.5

Guide Version 1.5.2

Released: June 2004

THE *Open* GROUP

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

### **Copyright 2003,2004 The Open Group**

All rights reserved. No part of this documentation may be reproduced, stored in retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as stated in the end-user license agreement, without the prior permission of the copyright owners. The text of the end-user license agreement appears in Appendix A of this document. A copy of the end-user license agreement is contained in the file License, which accompanies the TETware distribution.

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

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

Win 32<sup>TM</sup>, Windows NT<sup>TM</sup> and Windows 95<sup>TM</sup>, 98<sup>TM</sup> and 2000<sup>TM</sup> are registered trademarks or trademarks of Microsoft Corporation.

This document is produced by

The Open Group  
Apex Plaza  
Forbury Road  
Reading  
Berkshire  
RG1 1AX  
England.

# CONTENTS

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	<i>Preface.....</i>	<i>1</i>
1.2	<i>Audience .....</i>	<i>1</i>
1.3	<i>Conventions Used in this Guide .....</i>	<i>1</i>
1.4	<i>Related Documents.....</i>	<i>2</i>
1.5	<i>Problem Reporting .....</i>	<i>2</i>
<b>2</b>	<b>TETware GUI Overview.....</b>	<b>3</b>
2.1	<i>Architecture .....</i>	<i>3</i>
2.1.1	<i>License File .....</i>	<i>3</i>
2.2	<i>Test Run.....</i>	<i>4</i>
2.3	<i>Scenario File .....</i>	<i>4</i>
2.4	<i>The GUI.....</i>	<i>5</i>
2.4.1	<i>Tree Node Window .....</i>	<i>5</i>
2.4.2	<i>The Tree Nodes .....</i>	<i>6</i>
2.4.3	<i>The Tool Bar .....</i>	<i>8</i>
2.4.4	<i>Toolbar items.....</i>	<i>8</i>
2.4.5	<i>Menu Items.....</i>	<i>10</i>
2.4.6	<i>Keyboard Shortcuts .....</i>	<i>12</i>
2.5	<i>Dialog Boxes .....</i>	<i>13</i>
2.6	<i>Modes of Operation.....</i>	<i>13</i>
2.7	<i>Local, Remote and Distributed Testing.....</i>	<i>14</i>
2.8	<i>Embedded System Testing .....</i>	<i>14</i>
<b>3.</b>	<b>Installing the TETware GUI .....</b>	<b>15</b>
3.1	<i>Installation on Windows Systems .....</i>	<i>15</i>
3.2	<i>De-installation on Windows Systems.....</i>	<i>15</i>
3.3	<i>Installing on UNIX and Linux Systems.....</i>	<i>16</i>
3.4	<i>Configuring the Database Connection.....</i>	<i>16</i>
3.4.1	<i>Setup Function (Admin) (32-bit Windows Platform).....</i>	<i>16</i>
3.4.2	<i>Setup Function (Admin) (UNIX® and Linux® Operating Systems).....</i>	<i>17</i>
3.5	<i>Launching the TETware GUI.....</i>	<i>17</i>
3.6	<i>Help .....</i>	<i>18</i>
3.7	<i>Support .....</i>	<i>18</i>

# CONTENTS

<b>4.</b>	<b>Configuring the Environment .....</b>	<b>19</b>
4.1	<i>Configuring the Windows Environment .....</i>	19
4.2	<i>Configuring the UNIX or Linux Environment .....</i>	19
4.3	<i>Creating the /tmp directory.....</i>	19
4.4	<i>Configuring Distributed TETware .....</i>	20
<b>5.</b>	<b>Demonstrations.....</b>	<b>21</b>
5.1	<i>TETware Lite Demonstration.....</i>	21
5.1.1	Introduction .....	21
5.1.2	Opening the Test Run Files .....	22
5.1.3	Configuring the Demonstrations .....	22
5.1.4	Executing the Demonstration .....	24
5.1.5	Handling the Results .....	24
5.2	<i>TETware RT Demonstration.....</i>	25
5.2.1	Introduction .....	25
5.2.2	Selection the Test Run File.....	25
5.2.3	Using a Different Location.....	26
5.2.4	Selecting the IP Address.....	26
5.2.5	Running the Test Run.....	26
<b>6</b>	<b>Getting Started .....</b>	<b>27</b>
6.1	<i>Opening Test Run File(s).....</i>	27
6.1.1	Opening an Existing Test Run File(s) .....	27
6.1.2	Create a New Test Run.....	28
6.2	<i>Removing Test Run File(s). .....</i>	29
<b>7</b>	<b>Configuration.....</b>	<b>31</b>
7.1	<i>Main Configuration Setting.....</i>	31
7.1.1	Main Tab .....	32
7.1.2	Distributed Tab.....	33
7.1.3	Embedded Tab.....	34
7.1.4	Scenario Filter Tab .....	36
7.1.5	Custom Result Codes Tab .....	37
7.1.6	Alternate Config Files Tab .....	38
7.1.7	User Defined Variables Tab .....	39
7.1.8	Advanced Tab.....	41
7.2	<i>Local System Configuration .....</i>	42
7.2.1	Local: Environment Tab.....	43
7.2.2	Local: Build Mode.....	44
7.2.3	Local: Execute Mode.....	46

# CONTENTS

7.2.4	Local: Clean Mode .....	48
7.3	<i>Importing Configuration files</i> .....	50
7.3.1	User Defined Variables .....	51
7.3.2	Distributed config files .....	51
7.3.3	How to import a configuration file .....	51
<b>8</b>	<b>Scenario File(s) .....</b>	<b>53</b>
8.1	<i>What is a Scenario File?</i> .....	53
8.2	<i>Adding a Scenario File</i> .....	53
8.3	<i>Selecting Test Cases from a Scenario File</i> .....	54
8.4	<i>Editing a Scenario File</i> .....	55
8.5	<i>De-Selecting Test Cases from the Scenario File</i> .....	56
8.6	<i>Creating a Scenario File</i> .....	56
8.6.1	Including Test Cases in the Scenario File Creation (ADD).....	57
8.6.2	Removing Test Cases from Scenario File Creation (REMOVE).....	58
8.6.3	Resetting the Whole File Selection Process (RESET).....	59
8.6.4	Editing the Selected Variable (EDIT).....	59
8.6.5	Saving the Scenario File (CREATE).....	59
8.7	<i>Removing a Scenario File</i> .....	60
<b>9</b>	<b>Remote Systems (Distributed TETware only) .....</b>	<b>61</b>
9.1	<i>Adding a Remote System</i> .....	61
9.1.1	Using the Remote Wizard to Add a Remote System.....	61
9.1.2	Using the Tree Node or Menu Bar to Add a Remote System .....	62
9.2	<i>Configuration Settings on the Remote System</i> .....	62
9.2.1	Remote: Environment.....	63
9.2.2	Remote: Build Mode .....	65
9.2.3	Remote: Execute Mode .....	67
9.2.4	Remote: Clean Mode.....	69
9.3	<i>Importing Configuration Files</i> .....	71
9.3.1	Importing a Build Mode File.....	71
9.3.2	Importing an Execute Mode File .....	72
9.3.3	Importing a Clean Mode File .....	72
2.1.1	Importing a Distributed Config File .....	72
9.4	<i>Copying a Remote System</i> .....	73
9.5	<i>Removing a Remote System</i> .....	73
9.6	<i>Rename Remote System</i> .....	74
9.7	<i>Systems Files</i> .....	74

# CONTENTS

9.7.1	The Systems File.....	75
9.7.2	Makefiles .....	75
<b>10</b>	<b>Reporting – Configure Report Writer &amp; Database.....</b>	<b>77</b>
10.1	Configuring the TETware Report Writer .....	77
10.1.1	Creating a Report.....	81
10.2	Configuring the Database Extraction tool .....	81
10.2.1	Database Setup tab .....	81
10.2.2	Remote Database:.....	83
10.2.3	Extract Journal Directly.....	83
10.2.4	Database .....	83
10.3	Configuring your own Report Writer .....	85
<b>11</b>	<b>Executing a Test Run .....</b>	<b>89</b>
11.1	Pre-execution Criteria.....	89
11.2	Starting Execution of a Test Run.....	90
11.3	Execution Configuration Files .....	91
11.4	Quick Re-Run .....	93
11.5	Advanced Re-Run .....	94
11.5.1	Choosing Results Codes .....	94
11.5.2	Choosing Modes of Operation.....	95
11.6	Stop Test Run.....	96
11.7	Execution Output.....	97
11.7.1	Execution Output.....	97
11.7.2	Error Output .....	97
11.7.3	Journal File.....	97
11.8	Test Case Locking .....	98
<b>12</b>	<b>Journal.....</b>	<b>99</b>
12.1	Generating Journal Output .....	99
12.2	Viewing a Journal File.....	99
<b>13</b>	<b>Reporting.....</b>	<b>101</b>
13.1	Generating a Report with the TETware Report Writer .....	101
13.2	Database Extract.....	101
13.3	Generating your own Report:.....	102
<b>14</b>	<b>Saving a Test Run.....</b>	<b>103</b>

# CONTENTS

14.1	<i>Using the 'Save' option</i> .....	103
14.2	<i>Using the 'Save As' Option</i> .....	104
14.3	<i>Setting Defaults</i> .....	104
14.3.1	<i>Config menu default setting</i> .....	104
14.3.2	<i>Default setting via right-clicking on a tree node</i> .....	105
<b>15.</b>	<b>Wizards</b> .....	<b>107</b>
15.1	<i>Test Run Setup (Basic) Wizard</i> .....	107
15.2	<i>Test Run Setup (Advanced) Wizard</i> .....	108
15.3	<i>New Remote System Wizard</i> .....	109
<b>16</b>	<b>Help</b> .....	<b>111</b>
16.1	<i>TETware GUI Help Pages (In-Line Help)</i> .....	111
16.2	<i>TETworks Web Site (On-line Help)</i> .....	111
16.3	<i>How to use the Help Pages</i> .....	111
16.3.1	<i>Using the Tree</i> .....	112
16.3.2	<i>Closing the Help window</i> .....	112
16.4	<i>Creating Your Own Help Pages</i> .....	112
16.4.1	<i>Examples</i> .....	116
16.5	<i>Feedback and Problem Reporting</i> .....	117
<b>17</b>	<b>Exiting the TETware GUI</b> .....	<b>119</b>
17.1	<i>Exiting via the File menu</i> .....	119
17.2	<i>Using the system exit</i> .....	120
<b>Appendix A - TETware License</b> .....		<b>121</b>
<b>Appendix B - TETware GUI Database Design</b> .....		<b>125</b>
	<i>Why did we Create the 'Database Extract Tool'?</i> .....	125
	<i>What are its Features and Abilities?</i> .....	125
	<i>Database Tables</i> .....	126
	<i>Database Fields</i> .....	127

## LIST OF FIGURES

Figure 1: The TETware GUI.....	5
Figure 2: The Toolbar (undocked) .....	8
Figure 3: Main Configuration tab.....	32
Figure 4: Distributed tab.....	33
Figure 5: Embedded tab .....	35
Figure 6: Scenario Filter.....	36
Figure 7: Custom Result codes.....	38
Figure 8: Alternate Config Files.....	39
Figure 9: User Defined Variables.....	40
Figure 10: Advanced tab .....	41
Figure 11: Local: Environment tab .....	43
Figure 12: Local: Build Mode.....	45
Figure 13: Local: Execute Mode .....	47
Figure 14: Local: Clean Mode.....	50
Figure 15: Editing Scenario File .....	55
Figure 16: Create Scenario file Tab .....	57
Figure 17: Scenario File Chooser.....	58
Figure 18: Remote: Environment .....	64
Figure 19: Remote: Build Mode.....	65
Figure 20: Remote: Execute Mode.....	68
Figure 21: Remote: Clean Mode .....	70
Figure 22: TETware Report Writer .....	78
Figure 23: Database Setup tab.....	82
Figure 24: Database Tab.....	84
Figure 25: Generic Report Writer .....	86
Figure 26: Choose Results Codes.....	95
Figure 27: Choose modes of operation.....	95
Figure 28: Execution Output .....	98
Figure 29: Relationship between Database Tables.....	126

# LIST OF TABLES

Table 1: Tree Node Icons .....	7
Table 2: Toolbar Icons .....	9
Table 3: File Menu .....	10
Table 4: Run Menu.....	10
Table 5: Config Menu .....	11
Table 6: Import Menu.....	11
Table 7: Wizard Menu.....	11
Table 8: Help Menu.....	12
Table 9: Keyboard Shortcuts.....	12
Table 10: Test Run Setup (Basic) Wizard.....	108
Table 11: Test Run Setup (Advanced) Wizard.....	109
Table 12: New Remote System Wizard .....	110
Table 13: Help Icons .....	112
Table 14: Run of Test Run Table .....	127
Table 15: Configuration Variables Table.....	127
Table 16: Test Purpose Table.....	128



# 1 Introduction

## 1.1 Preface

This document is the TETware GUI User Guide.

TETware is a Test Execution Management System that takes care of the administration, reporting, and sequencing of the tests providing a single common user interface for all of the tests that you develop.

Both TETware and the TETware GUI have been tested and used on UNIX, Linux and Windows operating systems. The GUI includes all of the functionality of TETware and adds a number of extra features.

Throughout this document, the Windows NT, 2000, 2003, XP and 9x operating systems are referred to collectively as **Win32** systems. The individual names are only used when it is necessary to distinguish between them.

## 1.2 Audience

This document should be read by systems administrators installing TETware on their computer systems, and by software testing engineers who will use TETware from the GUI to run verification test suites.

## 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.

## 1.4 Related Documents

Refer to the following documents for additional information about the TETware:

- *TETware Installation Guides*
- *TETware Programmers Guide*
- *TETware Release Notes*
- *TETware User Guide*
- *TETware Knowledgebase*
- *TETware RT, Realtime and Embedded Systems Extension Guide*
- *TETware Report Writer User Guide*

You should read the Installation Guide thoroughly before attempting to install and use each new release of the TETware. Please refer to Sections 3 and 4 of this Guide for details of Installing and Configuring the TETware GUI.

## 1.5 Problem Reporting

If you have subscribed to TETware support and you encounter a problem while installing and using TETware, you can send a support request by electronic mail using the dedicated email address that is provided. Evaluators should email to [tetware\\_manager@opengroup.org](mailto:tetware_manager@opengroup.org)

All Problem Reports are welcome and actively encouraged. The more problems that are found and fixed the better the product will be. Please submit all bugs and queries found. Also, please submit requests for features and upgrades.

## 2 TETware GUI Overview

### 2.1 Architecture

The TETware GUI is written as a wrapper around a core. The wrapper is written purely in Java 2 using the full capabilities of the Swing APIs. Please note that the TETware GUI requires Java 2 (also known as JDK 1.2) or later.

The core (TETware) is written in C, and requires a C compiler.

TETware requires POSIX.1 conformance. It is supported on UNIX operating systems, Linux and 32-bit Windows Systems.

The system requirements for Windows Systems are:

- Visual C++ 4.0 or later
- MKS Toolkit for Windows NT Version 4.4 or later

Language bindings are available for the development of test cases in: C, C++, Java, Perl, Korn Shell, XPG3 Shell, POSIX Shell, Python and Tcl.

#### 2.1.1 License File

The TETware GUI requires the presence of a license file `license_file.txt` in order to operate. This file should be located in the same directory as the GUI. The license file also has a number of other functions:

- It contains the date on which the GUI license will expire. This feature is normally only used for Evaluation purposes.
- It controls the version of the GUI that is being used, the options are:
  - TETware Lite
  - Distributed TETware
  - TETware RT
  - OEM versions

If an OEM version is in use the license file may control the branding of the GUI, for example the image displayed in the Display Window (see Figure 1) need not refer to TETware.

## 2.2 Test Run

The TETware GUI uses the concept of a `Test Run`; this is a logical and conceptual unit that encompasses all the necessary data needed to ensure that a series of Test Cases can be executed successfully.

The `Test Run` contains a number of elements, some mandatory and others optional that when combined enable the User to concentrate more on the purpose behind the tests.

The `Test Run` contains configuration data for both local and, optionally, remote machines involved in testing. It includes a `Scenario` file that will be used to manipulate the tests. It also contains information relating to the operation of the TETware GUI including the advanced features that the TETware GUI offers.

Because the `Test Run` is a logical unit, it can be saved as one unit, meaning that all the data relating to a given execution of tests is saved as a whole. This means that the User no longer has to worry about the management of configuration files and environment variables.

## 2.3 Scenario File

A `scenario` file contains one or more test scenarios for a test suite. Each test scenario lists all the test cases and describes how they are to be processed.

When a `Test Run` is created, there will be no `Scenario` file included. This is shown at the bottom of the `Test Run` tree with the label: `No Scenario`. The user **MUST** add an existing scenario file, or create a new scenario file in order to run the `Test Run`. However, once that `Test Run` is saved, that scenario file is saved with the `Test Run`.

## 2.4 The GUI

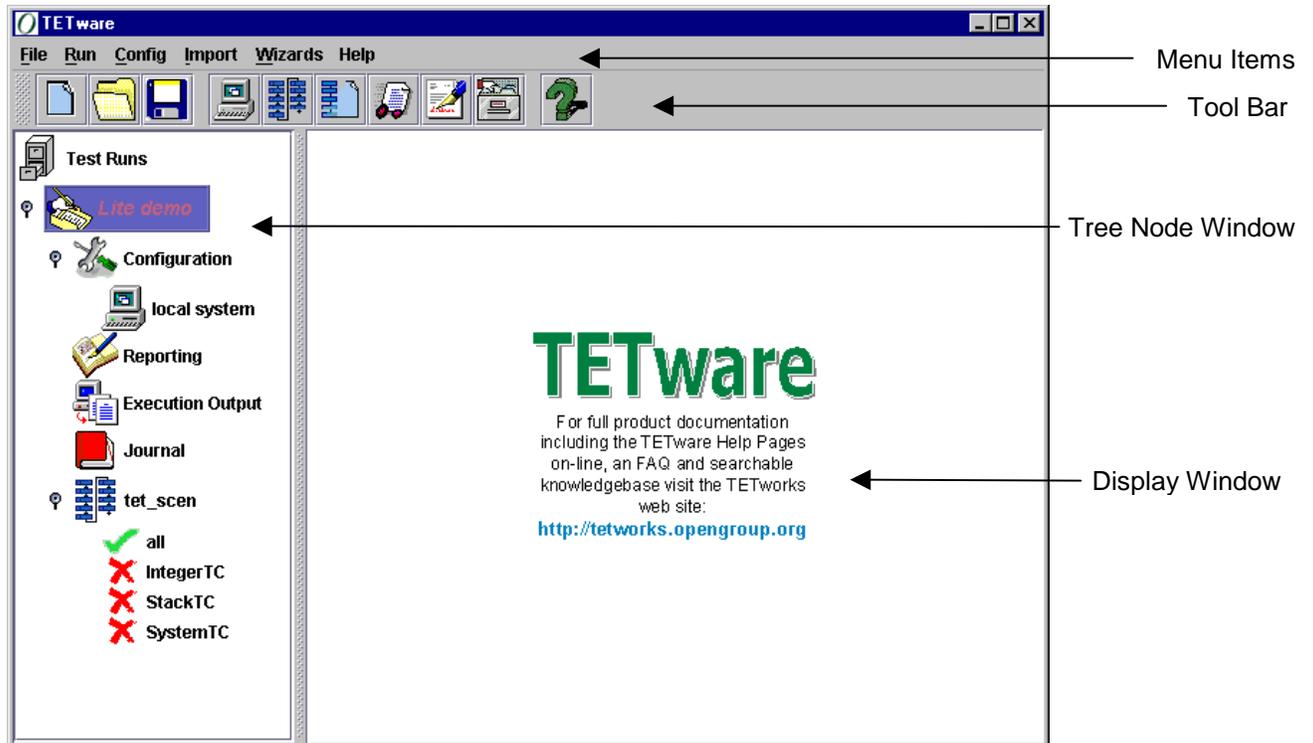


Figure 1: The TETware GUI

### 2.4.1 Tree Node Window

The Tree Node Window is where the Test Runs are listed. Since a number of Test Runs can be managed at any given time, the concept of an 'Active Test Run' is introduced. The Active Test Run is the Test Run that is currently being examined or manipulated by the User. The configuration screens, reporting node, execution node, journal node and scenario node will all display information relating to the Active Test Run.

The Active Test Run is indicated in the TETware GUI by the Test Run name being highlighted in **RED**.

You can select a Test Run to be 'Active' by clicking on the Test Run itself or any of its child nodes, so by clicking on the Scenario file you would make the Test Run associated with that Scenario file the Active Test Run. Once a selection is made, you can move up or down the tree by pressing the up or down arrow keys. Any node clicked on in the TETware GUI will be highlighted in **BLUE**.

Moving between Test Runs will cause the associated screens to change, dependent on the data associated with the Test Run chosen. At any given time the configuration screens will display the Active Test Runs configuration data.

Please note that expanding and contracting the tree nodes via the use of the folder handles will NOT cause the Active Test Run or selected node to change.

## 2.4.2 The Tree Nodes

Each Test Run is represented in the Test Run tree by a series of tree nodes, each of which give the User a 'window' into a different aspect of the Test Run and its data. Right clicking on most of these nodes provides the user with access to Test Run configuration options.

The main 'root' node that all Test Runs are added to is represented by the icon  .

When a Test Run is created or opened this node  is also opened.

The following table gives a description of each tree node and its usage within the Test Run.

Tree Node Icon	Tree Node Name	Description
	Test Run Open	This node is the base node for all the subsequent nodes relating to an individual Test Run, from this node you are able to add a scenario file, save and remove the Test Run. It is the starting point for the Test Run.
	Test Run Closed	This node is the closed version of the Test Run Open node described above; when this icon is displayed it indicates that the Test Run has been hidden in the tree.
	Configuration	This node is starting point for the configuration for the Test Run. Attached to this node is a 'local system' node, which holds the local systems configuration. Remote systems are also attached to this node. If you click on this node then the Main configuration screen for the Test Run will be displayed.
	Local System	This node represents the local system configuration. The local system is the system on which TETware will be invoked. Clicking on this node displays the local system configuration screens.
	Remote System	This node represents a remote system that will be used in the Test Run. This node is optional and is only required when TETware is being used in distributed mode, i.e. for remote or distributed testing. Clicking on this node displays the configuration screens for that remote system.
	Reporting	Clicking on this node will take you to the Reporting screen where you can specify a report to be generated using a Journal file or configure the database extract tool and specify the Journal file parameters to be passed to the database.
	Execution Output Started	As the name suggests this node, when shown, indicates that the Test Run is currently executing. The output can be viewed in the Execution screens. The Execution screens can be seen by clicking on this node. This icon will be displayed for as long as the Test Run is executing.
	Execution Output Stopped	As the name suggests this node, when shown, indicates that the Test Run is NOT currently executing. Clicking on this icon will display the Execution screens.
	Journal	Clicking on this icon will take you to the Journal file display screen. If you have performed a Test Run for which there was a Journal file generated then it will be displayed here. Right clicking on this node allows you to Browse the file system for a Journal file. <i>Note: you cannot open a Journal file that is greater than 2MB in size.</i>
	Scenario File	This node represents the Scenario file. If valid scenarios exist within the Scenario file then these are displayed as scenario nodes below this node. Clicking on this node will display the Scenario file in the Scenario file display window. Right clicking on this node allows you to refresh the scenario file (basically re-read the Scenario file if it has changed), include and exclude all the scenarios, remove the Scenario file, and display the path to the Scenario file. If no scenario file is included then right clicking on this node allows you to add an existing scenario file, or create a new scenario file.
	Scenario Included	This node represents a scenario from the Scenario file that should be Included when the Test Run is executed. Single clicking on the scenario node will toggle its included/excluded status.
	Scenario Excluded	This node represents a scenario from the Scenario file that should be Excluded when the Test Run is executed. Single clicking on the scenario node will toggle its included/excluded status

Table 1: Tree Node Icons

### 2.4.3 The Tool Bar

The Tool Bar, shown below, is provided to allow the User access to certain frequently used functions.



Figure 2: The Toolbar (undocked)

It is possible to 'dock' the toolbar into any of the edges of the GUI.

It is also possible to have the toolbar floating in its own frame, as in Figure 2, this gives the User maximum flexibility as to the positioning of the toolbar. To undock the toolbar click on the area to the left of the New Test Run icon  and drag it to the new position while holding the left mouse button down.

Note that the Toolbar in Figure 2 is the one that is displayed when the GUI is used with Distributed TETware. The Remote System icon  is displayed. This icon is not displayed when the GUI is used with TETware Lite or TETware RT.

### 2.4.4 Toolbar items

Table 2 describes the icons on the toolbar and their function:

Icon	Name	Description
	New Test Run	Create a new blank Test Run; the User will be prompted for the Test Run name.
	Open an existing Test Run	This will display a file selection dialog that the User can browse to open an existing Test Run, a file filter of *.trf is provided to find Test Run files quickly and easily.
	Save the Active Test Run	This button will perform the same function as the 'Save' function in the 'File' menu. Note: this button will only work when there is an active Test Run.
	Start New Test Run Wizard	This button starts the Test Run Setup (Basic) Wizard. See Section 15.1.
	Start New Remote System Wizard	This button starts the New Remote System Wizard. See Section 15.3.
	Add a Scenario file	When a test run is active, this button opens a file chooser box and prompts the user to select a Scenario File. The selected file is then added to the active Test Run. If there is no active test run an error dialog box is displayed.
	Create a Scenario file	When a test run is active, this button takes the user to the Create Scenario File tab.
	Execute Test Run	This button will start the execution of the active Test Run. The user is prompted for confirmation before the Test Run executes.
	Run the TETware Report Writer	This button generates a Report using the TETware Report Writer using the Journal file specified in the TETware Report Writer tab.
	Start Extract to Database	This button starts the process of extracting Journal file parameters and passing them to the database specified on the Database Setup tab.
	Help	This button launches the TETware GUI Help pages.

Table 2: Toolbar Icons

## 2.4.5 Menu Items

The Menu items can be selected by clicking on them or using the keyboard shortcut (Alt-\*) where \* is the underscored letter.

### 2.4.5.1 File Menu

Menu item	Description	See Section
<u>N</u> ew Test Run	Create a new Test Run.	6.1.2
<u>O</u> pen Test Run	Open an existing Test Run.	6.1.1
<u>S</u> ave Test Run	Save the currently Active Test Run.	14.1
Save <u>A</u> s Test Run	Save the currently Active Test Run as a different name.	14.2
Remove Test Run	Remove the currently Active Test Run.	6.2
<u>E</u> xit	Exit the TETware GUI	17

**Table 3: File Menu**

### 2.4.5.2 Run Menu

Menu item	Description	See Section
<u>E</u> xecute Test Run	Executes the currently Active Test Run.	11.2
Quick Re-run	Re-execute a Test Run for all Tests that have a result code of FAILED and/or UNTESTED but only for the Modes of Operation used for the initial execution.	11.4
Advanced Re-run	Re-execute a Test Run based upon the criteria specified by the User.	11.5
<u>S</u> top Test Run	Stops a Test Run executing (if already executing).	11.6

**Table 4: Run Menu**

### 2.4.5.3 Config Menu

Menu item	Description	See Section
Add Scenario File	Add a Scenario File to the currently Active Test Run (if one is not already present).	8.2
Remove Scenario File	Remove the Scenario File from the currently Active Test Run (if one is already present).	8.7
Create Scenario File	Create a Scenario File and add it to the currently Active Test Run	8.6
Add Remote System	Add a Remote System to the currently Active Test Run.	9.1
Save as default settings	Save the currently Active Test Runs main and local system configuration as the defaults.	14.3
Revert to default settings	Set the currently Active Test Runs main and local system configuration to the defaults.	14.3

**Table 5: Config Menu**

### 2.4.5.4 Import Menu

Menu item	Description	See Section
Import Build mode file	Import a Build mode file into the currently Active Test Run.	9.3.1
Import Execute mode file	Import an Execute mode file into the currently Active Test Run.	9.3.2
Import Clean mode file	Import a Clean mode file into the currently Active Test Run.	9.3.3
Import Distributed config file	Import a Distributed config file into the currently Active Test Run.	9.3.4

**Table 6: Import Menu**

### 2.4.5.5 Wizard Menu

Menu item	Description	See Section
Test Run Setup (Basic) Wizard	Steps the User through the four basic steps that must be carried out to configure a new Test Run	15.1
Test Run Setup (Advanced) Wizard	Steps the User through all the steps that might be carried out to configure a new Test Run.	15.2
Start New Remote System Wizard	Steps the User through all the steps that might be carried out to configure a new Remote System	15.3

**Table 7: Wizard Menu**

### 2.4.5.6 Help Menu

Menu item	Description	See Section
About the TETware GUI	Display information relating to the TETware GUI	(none)
Browse <u>H</u> elp pages	Browse these help pages.	16.1

**Table 8: Help Menu**

### 2.4.6 Keyboard Shortcuts

The menus, and some menu items, can also be accessed by using keyboard shortcuts as follows:

Shortcut	Function	See Section
Alt-A	Save <u>A</u> s Test Run	14.2
Alt-B	<u>B</u> asic (Test Run Setup) Wizard	15.1
Alt-C	<u>C</u> onfiguration Menu	2.4.5.3
Alt-D	<u>D</u> atabase Extract	13.2
Alt-E	<u>E</u> xecute Test Run	11.2
Alt-F	<u>F</u> ile Menu	2.4.5.1
Alt-G	<u>G</u> enerate Report	13.1
Alt-H	Browse <u>H</u> elp Pages	16.1
Alt-I	<u>I</u> mport Menu	2.4.5.4
Alt-K	Create a Scenario File	8.6
Alt-L	Add a Scenario File	8.2
Alt-M	<u>R</u> e <u>M</u> ote System Wizard	9.1
Alt-N	<u>N</u> ew Test Run	6.1.2
Alt-O	<u>O</u> pen Test Run	6.1.1
Alt-Q	<u>Q</u> uick Re-run	11.4
Alt-R	<u>R</u> un Menu	2.4.5.2
Alt-S	<u>S</u> ave Test Run	14.1
Alt-V	<u>A</u> d <u>V</u> anced Re-run	11.5
Alt-W	<u>W</u> izard Menu	2.4.5.5
Alt-X	<u>E</u> Xit	17
Alt-Z	Advanced (Test Run Setup) <u>W</u> i <u>Z</u> ard	15.2

**Table 9: Keyboard Shortcuts**

## 2.5 Dialog Boxes

The following buttons are general to most of the dialog boxes and tabs found within the GUI:

<b>BROWSE</b>	Clicking the <code>Browse</code> button will display a file selection box, which can be used to browse the file system to make selections.
<b>CLEAR</b>	clear the field you want to enter a variable into or clear whatever you have previously entered
<b>CANCEL</b>	aborts the present operation entirely.
<b>BACK</b>	go back one step to the previous field.
<b>NEXT</b>	move to the next field.
<b>FINISH</b>	finish with the settings of the variable, save your settings and exit.
<b>ADD</b>	adds a variable, in an input field, to a list.
<b>REMOVE</b>	removes a variable from a list. To remove multiple entries you can hold the Shift or Ctrl key and select rows in the list, if you hold 'Shift' then you can select a contiguous list of Test Cases, if you hold Ctrl you can choose a dis-contiguous list of Test Cases.
<b>RESET</b>	removes all the variables from a list.
<b>EDIT</b>	removes a variable from a list and places it in the input field for editing. Note the Add button must be used to return it to the list.

## 2.6 Modes of Operation

The TETware GUI operates in one of several modes as follows:

<b>build mode</b>	in this mode, test cases are built or compiled
<b>execute mode</b>	in this mode, Test Cases are run.

**clean modes** in this mode, test cases are cleaned; i.e., the test case directory is returned to the state that it was in before the test case was built and/or executed. e.g. all files created are deleted.

One or more modes of operation are specified by command-line options each time that the TETware GUI is invoked.

## 2.7 Local, Remote and Distributed Testing

The **Lite** version of TETware is able to process non-distributed test cases on the Local System.

The **Distributed** version of TETware is able to process non-distributed test cases on both the Local System and on one or more Remote Systems. In addition, it is able to process distributed test cases where parts of each test case execute on the Local System and/or on one or more Remote Systems.

Distributed systems are identified by a three-digit system ID. System ID `zero` (000) refers to the Local System and other system ID values refer to Remote Systems. In this context, the Local System is defined as the one on which TETware is invoked. Usually, different TETware systems are located on different physical machines. However, there is no reason why different system IDs should not refer to the same machine.

## 2.8 Embedded System Testing

TETware RT is able to control the execution of tests on an Embedded POSIX Realtime System that supports one of the profiles defined in POSIX 1003.13. For detailed information about Embedded Realtime Systems, please refer to the TETware RT, Realtime and Embedded Systems Extension Guide.

## 3. Installing the TETware GUI

**IMPORTANT:** The TETware GUI requires Java 2 (also known as JDK 1.2).

### 3.1 Installation on Windows Systems

**Note:** Windows users MUST set the path to their JDK in their AUTOEXEC.BAT file or equivalent

The TETware GUI can be installed on 32-bit Windows platforms using a zip file:

- to install extract the zip file and then create a shortcut to TETware.jar

**Note:** To successfully install TETware you should NOT have spaces in the directory path of TET\_ROOT. Therefore TETware we recommend it be installed and the demo test suite assume it to be installed into C:\TETWARE not C:\Program Files.

### 3.2 De-installation on Windows Systems

To remove TETware files from your machine:

1. Remove the C:\TETWARE directory and its contents.

### 3.3 Installing on UNIX and Linux Systems

If you are working in a UNIX or Linux environment, you install the TETware GUI from a tar file. In this case, you need to select the tar file appropriate for the *Type of Testing* (Distributed or Local) and the *Operating System* that you wish to use for your evaluation. For evaluators, the available tar file is a GNU Zip'd archive.

After downloading the appropriate tar file(s) please type:

```
gunzip <tar file>.tar.gz to uncompress the archive
tar xvf <tar file>.tar to install the TETware GUI.
```

### 3.4 Configuring the Database Connection

Before you can extract journal file parameters to a database, the following procedures must be carried out to establish a connection with the database. It should only be necessary to carry this function out once for each machine on which TETware is installed.

#### 3.4.1 Setup Function (Admin) (32-bit Windows Platform)

1. Click the Start button on your Windows Task Bar (Windows 9x and NT users). Then select Settings, then Control Panel. (If you are using Windows 2000 click on Administrative Tools). Double click on the ODBC Data Source (32-bit) icon.
2. Select the System DSN tab, and then click on the Add button.
3. Choose the driver for your data source (*e.g.* Microsoft Access Database Driver). Then click on Finish.
4. The ODBC Microsoft Access Setup window will open. Type in the Data Source Name (*e.g.* TETbase) then type its description (*e.g.* Microsoft Access Driver [\*].mdb). Then Click on either Select or Create.
5. If you clicked Select, a dialog box captioned 'Select Database' will be displayed, look for your file in the drive and the directory where you already have your Database set, then click on OK.

6. If you picked `Create`, a dialog box captioned 'New Database' then chooses the drive and directory where you have your Database file. (We suggest you use the same directory where you have your TETware program installed, *i.e.* your `TET_ROOT` directory). Then give the file a name and click on `OK`.
7. This will return you to the Setup Screen. Now click on `OK`.
8. An ODBC Microsoft Access message box will open, with a message "Database `Z:\ddd\eee.mbd` was successfully created" (where 'Z' is the Drive letter, 'ddd' is the directory, and `eee.mbd` is the name of the Database file). Then click on `OK`.
9. Now close the entire ODBC Setup windows by clicking on `OK` or `Exit` all the way through.

### 3.4.2 Setup Function (Admin) (UNIX<sup>®</sup> and Linux<sup>®</sup> Operating Systems)

Unlike the Windows platforms, there is no need to go through any Database set up on either UNIX or Linux Operating Systems. The user will perform the only Database setup that needs to be carried out while configuring a `Test Run`, (please refer to Section 10.2).

## 3.5 Launching the TETware GUI

**Note:** Before you can run the TETware GUI you must have installed the `license_file.txt` file provided into the same directory that the `TETware.jar` file is installed.

To launch the TETware GUI go to the directory that contains `TETware.jar` file and use the command (note this is case sensitive)

```
java -jar TETware.jar
```

Windows users may double click on the TETware shortcut icon installed on their desktop (if created earlier), or on the `TETware.jar` file in the `gui` directory from within Windows Explorer. (There is a TETware icon in the same directory as the `TETware.jar` file that may be used for a desktop shortcut).

## **3.6 Help**

The TETware GUI Help Files can be viewed from within the TETware GUI .

## **3.7 Support**

If you have subscribed to TETware support and you encounter a problem while installing and using the TETware GUI, you can send a support request by electronic mail using the dedicated email address that is provided. Evaluators should email [tetware\\_manager@opengroup.org](mailto:tetware_manager@opengroup.org) .

## 4. Configuring the Environment

If you are planning to run Local tests using TETware Lite you should refer to Sections 4.1, 4.2 and 4.3 below. If you are planning to run Remote or Distributed tests using Distributed TETware then you should also refer to all Section 4.4.

### 4.1 Configuring the Windows Environment

Make sure that the setting of your `PATH` environment variable includes `tet-root\bin`.

Set your `TET_ROOT` environment variable to refer to `tet-root`. To do this, open up your `AUTOEXEC.BAT` and include a line as the example below:

```
PATH = C\tet-root\bin
```

### 4.2 Configuring the UNIX or Linux Environment

Make sure that the setting of your `PATH` environment variable includes `tet-root/bin`.

Set your `TET_ROOT` environment variable to refer to `tet-root`, Please open up your `.bat` or `.profile` file and include a line as the example below:

```
PATH = /tet-root/bin
```

### 4.3 Creating the `/tmp` directory

You need to specify a temporary directory that should be used for files generated while executing a test suite either on the Local system or on the Remote system(s). For example

```
/tet-root/tmp
```

## 4.4 Configuring Distributed TETware

If you are planning to run Remote or Distributed tests, you must have installed Distributed TETware, you now must configure **each system** that will participate in your tests. Please refer to Chapter 4 in the TETware Installation Guides for Instructions on how to do this.

## 5. Demonstrations

### 5.1 TETware Lite Demonstration

Once you have installed the TETware GUI you may wish to run the Lite Demonstration to test its operation. You must have installed TETware Lite to run this demonstration.

This demonstration uses tests written in Java.

#### 5.1.1 Introduction

This Demonstration provides a quick and easy way to present the capabilities of TETware and the TETware GUI when carrying out local testing. It forms the basis for the examples elsewhere in this User Guide. If you encounter difficulties this demonstration please refer, in the first instance, to the relevant part of this User Guide for help.

The Lite Demonstration comprises of a `Test Run` file and a `scenario` file that are used to manage the execution of three Java tests cases (provided):

<code>IntegerTC</code>	This is a Test Case Class to test <code>java.lang.Integer</code> . It tests to see if <code>intValue</code> , <code>toString</code> and <code>parseInt</code> work according to their specification in the Java Documentation.
<code>StackTC</code>	This is a Test Case Class to test <code>java.util.stack</code> , It tests the <code>java.util.stack</code> to ensure that the methods <code>pop</code> , <code>push</code> and <code>search</code> work according to their specification in the Java Documentation.
<code>SystemTC</code>	This is a Test Case Class to test <code>java.lang.system</code> . It tests to see if Java method 'exit' of the <code>java.lang.system</code> correctly terminates the running Java process and returns the given status code.

You may vary the selection of tests during the demonstration.

## 5.1.2 Opening the Test Run Files

Once you have launched the TETware GUI, you must open the appropriate Lite Demonstration Test Run File that is provided with the TETware GUI.

To do this use the "Open Test Run" item in the File menu (for alternative ways of opening a Test Run file please see Section 6.1).

Users will then be presented with a file selection dialog box that allows them to browse until the appropriate test run file (\*.trf) is found it should be in your TET\_ROOT/gui directory (i.e. the gui directory under the directory in which you installed TETware):

For UNIX users:       Select the file: `lite_java_unix.trf`

For Windows Users:    Select the file: `lite_java_win.trf`

Single click to select the file so that it appears in the File Name box and click Open. The Lite\_Demo Test Run will then appear as the first node in the Test Runs tree. Double click on Lite\_Demo to open the Test Run tree. If the message "No scenario file selected" appears please follow the instructions under item 7 in Section 5.1.3 below.

## 5.1.3 Configuring the Demonstrations

If you have accepted the default directory names when you installed TETware you may now proceed to Section 5.1.4, but If you changed the path to TETware components, you must now re-configure the Test Run before you can execute it. To do this there are a couple of changes required:

### 5.1.3.1 The Test Run Files

Note that the Test Run Files are in XML format and can be edited using a text editor or overridden in the GUI as in the next section. You should select the appropriate "trf" file for your platform (UNIX or Windows) and update the paths within that file appropriately. Exit the TETware GUI and then restart the GUI and reload the "trf" file.

### 5.1.3.2 The Local System Tab:

1. Underneath the `Lite_Demo` Test Run double click on `Configuration`, which expands the icon to add another graphic and then single click on `Local System`.
2. Next, click on the `Local: Environment` tab. In the `TET_ROOT` box at the top of this tab, you must specify the location of TETware on your system. If you changed the drive or directory in which this software is installed you must now select the `Browse` button. This opens a `Set value of: TET_ROOT` message box. Select the directory in which you instructed the installer to place the software.
3. Next click on the `Local: Build Mode` tab. On this tab, you must specify the location of the Build Tool for this demonstration in the `TET_BUILD_TOOL` box. The tool is called `jet-build.exe`. It is located in the `bin` directory under `TET_ROOT`. Now browse and select it.
4. Next click on the `Local: Execute Mode` tab. On this tab, you must specify the location of the Execute Tool for this demonstration in the `TET_EXECUTE_TOOL` box. The tool is called `jet-exec.exe`. It is located in the `bin` directory under `TET_ROOT`. Now browse and select it.
5. Next click on the `Local: Clean Mode` tab. On this tab, you must specify the location of the Clean Tool for this demonstration in the `TET_CLEAN_TOOL` box. The tool is called `jet-clean.exe`. It is located in the `bin` directory under `TET_ROOT`. Now browse and select it.
6. Next click on `Reporting` icon in the tree, and then click on `TETware Report Writer` tab. Go to the `Output File` box and specify the location and name of the file that you write the report to. Note that it is important that the file have the suffix “.html”.
7. Next, right click on the `No scenario` text at the bottom of the tree. Select the `Add Scenario File` option. The `Chose Scenario File` message box will open at the `TET_ROOT` level. Double click on the directory `jdemo` to open it, and then double click on the file `tet_scen` to open it. (**Note:**This file ‘`tet_scen`’ is editable, it is important that you do not edit it or save it, as this may mean that the demo will not execute.)
8. Double click on `tet_scen`, in the tree, and select at least one of the scenarios listed. Once selected the `red cross` will change to a green tick. You must have at least one green tick visible before you can execute the Test Run.

## 5.1.4 Executing the Demonstration

Press the Execute Test Run button  on the toolbar. A Confirm execution start message box opens. Select Yes to proceed. A no-entry sign  will appear on the Execution Output Tree Node and will remain there as long as the tests are executing. This no-entry sign will be automatically removed as soon as the test is finished. The demonstration test suite will be built, executed and cleaned up.

If you have text in the Error Output window please copy it into an email and send it to [tetware\\_manager@opengroup.org](mailto:tetware_manager@opengroup.org).

## 5.1.5 Handling the Results

### 5.1.5.1 View the Journal File

If a Test Run has been executed and a Journal file created by TETware then you can view the text of the Journal File that was generated by clicking on the Journal node . The Journal file will be parsed and placed into the "Journal Text" area. This behavior is automatic. You can now view the text of the Journal File that was generated when you executed the Test Run. Look at the first line of the Journal. It gives you the time and date that the Test Run was executed, (the date is in the format `yyyymmdd`) so you can confirm that it is your Test Run.

### 5.1.5.2 Report Writer

Click on the Reporting tab, and then TETware Report Writer. You should see that the Journal name has been updated to the file that you have just generated.

Select the Run TETware Report Writer button  on the toolbar. Using your browser you can locate and view the output file. It will be in the location that you specified while carrying out item 6 in Section 5.1.3.

### 5.1.5.3 Database

Select the `Start Extract to Database` button  on the toolbar. You should see that the directory specified at `Look in` is the directory number for your Journal file. Click on `Open`. Once the extraction is finished a `Database Extract finished` message box will open. Click on `OK` to close it. Now go and find the Journal file parameters in your Database.

## 5.2 TETware RT Demonstration

### 5.2.1 Introduction

This Section describes how to run the Embedded Demonstration that is supplied with TETware RT.

The demonstration is designed to be used on the sockets implementation of TETware RT.

First follow the instructions contained within the TETware RT, Realtime and Embedded Systems Extension Guide.

It is recommended that you install TETware and the Embedded Module into `/usr/local` for this demonstration, but if this is not possible, instructions are given on how to change the Embedded Demonstration to use a different location.

### 5.2.2 Selection the Test Run File

Once the installation of the Embedded Module is complete, click on the `Open Test Run` icon and select `embedded_demo.trf`.

### 5.2.3 Using a Different Location

This Test Run File presumes that TETware has been installed into `/usr/local`. If this is not the case various options will need to be changed.

In Configuration: Local System: Local Environment:

- `TET_ROOT` should point to top hierarchy in which TETware is installed.
- `TET_SUITE_ROOT` should point to the location of the `embedded_demo` directory

In Configuration: Local System: Local Execute:

- `TET_EXEC_TOOL` should point to the location of the binary “`tet3rtm`”.

The scenario file will also need to be opened. This can be done by using the menu `Config: Add Scenario File`. The scenario file can be found in the `embedded_demo` directory and is called `tet_scen`.

### 5.2.4 Selecting the IP Address

The IP address of the embedded system also needs to be set. This is held in the `Configuration: Local System: Local Embedded: TET3RT_RTSYS_ID`. Currently this is set to the loop back address in `127.0.0.1`.

### 5.2.5 Running the Test Run

Once the Test Run has been set up, it can be run by clicking on the Execute Test Run icon on the

tool bar . See Sections 5.1.4 and 5.1.5 for more information on Test Run Execution and Result Handling.

## 6 Getting Started

### 6.1 Opening Test Run File(s)

Once the TETware GUI has finished loading and the main screen is open, you are able to either:

- Open an existing Test Run.
- Create a New Test Run.

#### 6.1.1 Opening an Existing Test Run File(s)

An existing Test Run File is one that has been created at some point in the past and may be in any state, i.e. the configuration may be set or there may be a scenario file added or a combination of the two.

There are a number of ways to open an existing Test Run:

- Use the  icon in the toolbar.
- Use the `Open Test Run` item in the File menu.
- Right click on the `Test Runs` tree node and choose `Open Test Run`.
- Use keyboard shortcut `Alt-O`.

Whichever method you choose you will then be presented with a file selection dialog box captioned `Open Test Run`. All saved Test Run files are given a `.trf` file extension. In the file selection dialog box there is a file filter for `.trf` files. This allows easy access to saved Test Run files.

- Find the desired Test Run file.
- Select the Test Run file.
- Click `Open` to open the Test Run file.
- If you choose `Cancel`, the operation is cancelled and the file dialog box will be closed.

Once you have clicked `Open` and the file is a valid Test Run file, the Test Run will be opened and the Test Run added to the `Test Run's` tree node.

If you open a Test Run, when a Test Run with the same Test Run name is already open, a dialog box will be shown to ask if you really want to rename the nominated Test Run in order to open it. If the answer is YES a new dialog box will be displayed, to ask user for the new Test Run name, and the Test Run with the new Test Run name will be opened. If the answer is NO the operation will be cancelled.

### Invalid Test Run File

If you choose to open a file that is not a valid Test Run file then the Load Execution problem dialog box will be displayed with the message: 'The System cannot find the file Specified' and the file will NOT be opened.

Click on OK, and another dialog box will be displayed with the message 'Error: Test Run File [*file\_name*] not valid'. Click on OK again. Then you will need to select a new Test Run file.

## 6.1.2 Create a New Test Run

A new Test Run is one that is created from scratch. The User defines everything from the configuration setting to the scenario selection.

There are a number of ways to create a New Test Run:

- Use the  icon in the toolbar.
- Use the New Test Run item in the File menu.
- Right click on the Test Run tree node and choose New Test Run.
- Use the keyboard shortcut Alt-N.

There are also Basic And Advanced Wizards for setting up a new Test Run, please see Section 15.

Whichever method you choose to create a New Test Run you will then be prompted, by a dialog box, to specify the name of the Test Run. The name can be any string of characters.

**Please note:** Test Run names MUST be unique. This is to allow tracking of the Test Runs.

Once you click on OK, a check is made to see that the Test Run name does not already exist. If the Test Run name specified does not already exist then a new Test Run is created and added to the

Test Runs tree node. If the Test Run name does exist, then an error message is displayed, Click on OK to close the dialog box and choose another Test Run name.

## 6.2 Removing Test Run File(s).

To remove a Test Run from the tree you may follow any of the three methods below:

- Use the Remove Test Run menu item in the File menu.
- Right-click on the Test Run node in the tree and click the Remove Test Run menu item.
- Press the Delete key when the Test Run is highlighted.

Whichever way is chosen to remove the Test Run you will be prompted with a dialog box which ask you to Confirm removal of Test Run [*Name of Test Run*].

If the Yes button is clicked then all components of the Test Run will be removed from the tree. If No is chosen then the Test Run will not be removed and the process will be abandoned.



## 7 Configuration

The configuration Tree Node holds configuration variable assignments for each of TETware's modes of operation. In addition, configuration files may hold configuration variable assignments for use by test cases and by build and clean tools.

Each Tree Node has its own specific configuration settings.

- Main configuration.
- Local System Configuration.
- Remote System Configuration (see Section 9) (Distributed TETware only).

### 7.1 Main Configuration Setting

To access the Configuration screen you need to click on Configuration in the Tree Node, or press the up or down arrow keys once a tree node is highlighted.

Each Test Run must have one or more configuration files associated with it. These files contain configuration variable assignments, which are specified by the test suite author. When any test case is processed in a particular mode of operation, it reads variables from the configuration file for that mode.

There are a maximum of 7 tabs that can be accessed on the Configuration Screen. Which of the following tabs is displayed will depend on which version of TETware you are using: TETware Lite; Distributed TETware; or TETware RT.

- Main tab
- Distributed tab
- Embedded tab
- Scenario Filter tab
- Custom Result Codes tab
- Alternate Config files tab
- User Defined Variables tab
- Advanced tab

These tabs will now be described in detail.

### 7.1.1 Main Tab

The Main tab Configuration screen is displayed in Figure 3 below:

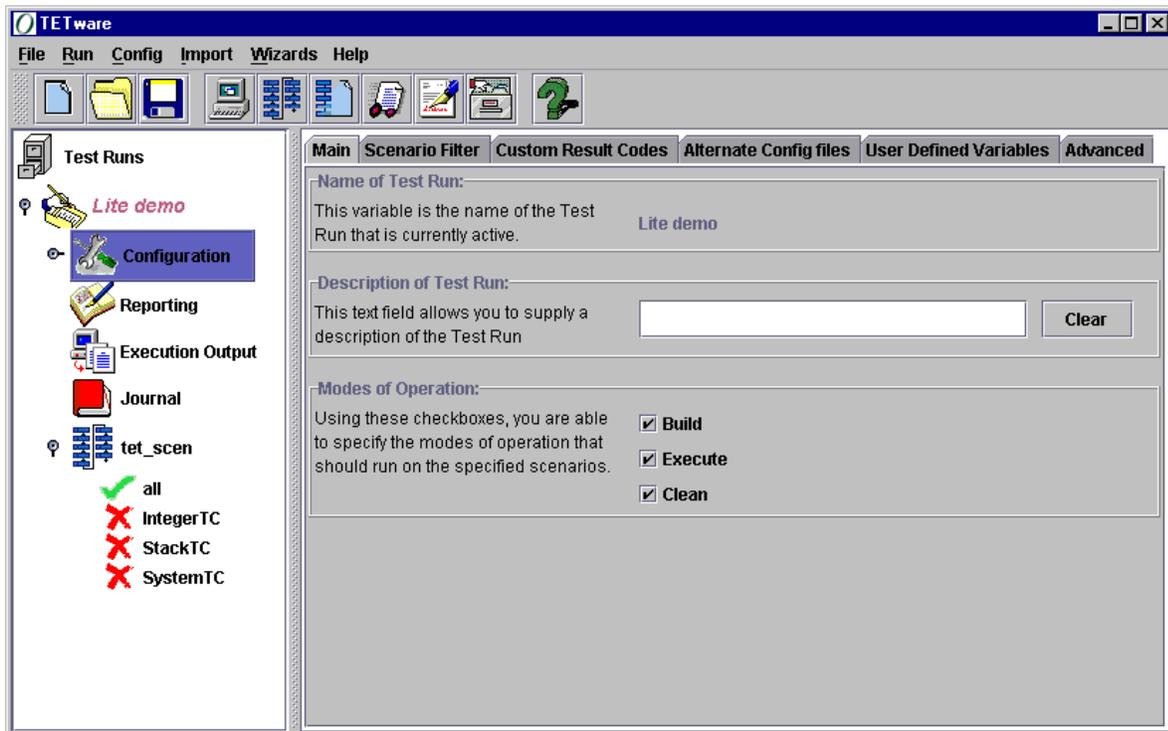


Figure 3: Main Configuration tab

This screen is broken down into a number of areas that can be chosen independently of the others.

The **Main** configuration tab holds the following values:

- |                                |  |
|--------------------------------|--|
| <b>Name of Test Run</b>        | As the name indicates, this field holds the name of the Test Run that is currently active. It is the name given to it by the user who first created the Test Run. It cannot be changed.  |
| <b>Description of Test Run</b> | This field holds a textual description of the Test Run, You are able to provide a 'one line' description that will give more information about a Test Run and it's purpose. You can edit the field by clicking in the box and editing or clear the contents using the <b>Clear</b> button. |
| <b>Modes of Operation</b>      | This list of checkboxes allows you to choose the modes of operation that you wish the TETware GUI to use. Any combination of modes is valid although to successfully execute the TETware GUI, you <b>MUST</b> choose at least one mode.  |

## 7.1.2 Distributed Tab

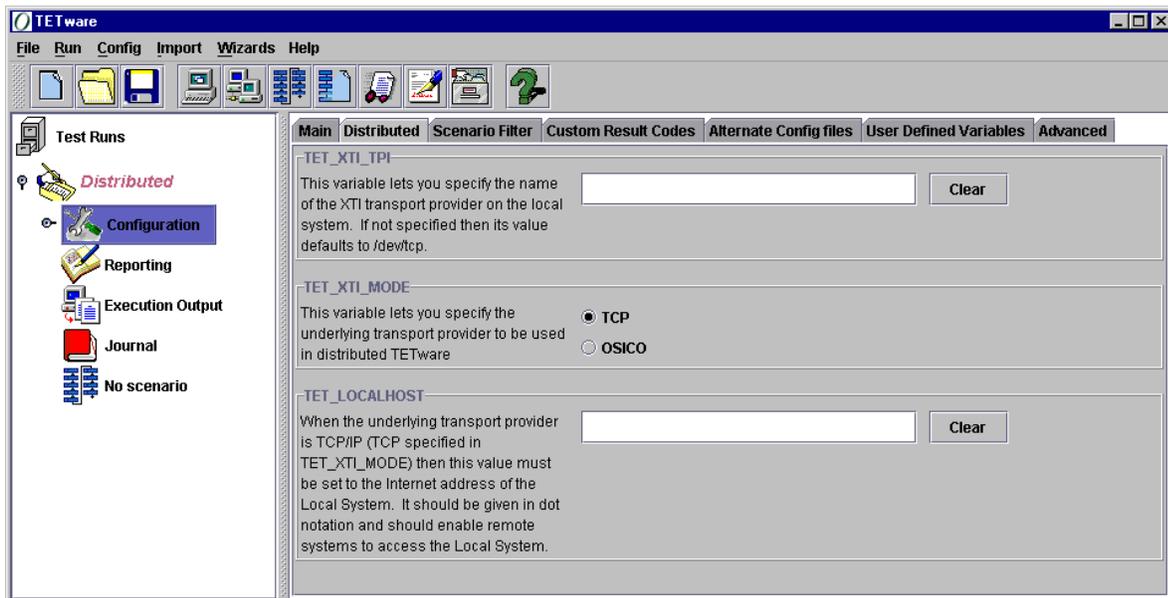


Figure 4: Distributed tab

If Distributed TETware is used then it may be that information regarding the Local Systems network address and underlying network protocol is required.

**TET\_XTI\_TPI** This variable lets you specify the name of the XTI transport provider on the Local System. If this is not specified, the value defaults to `/dev/tcp`.

**TET\_XTI\_MODE** This variable lets you specify the underlying transport provider to be used in TETware. There are two choices, either `TCP` or `OSICO`.

**TET\_LOCALHOST** When the underlying transport provider is TCP/IP (TCP specified in `TET_XTI_MODE`) then this value must be sent to the Internet address of the Local System. It should be given in dot notation and should enable the Remote Systems to access the Local System.

### 7.1.3 Embedded Tab

The configuration variables are used by TETware RT are:

**TET3RT\_RTSYS\_ID** Specifies the string that the Test Manager passes to Manufacturer-specific functions that take an `rtsys_id` argument. Whether or not the MSS actually uses it depends on the implementation. This variable **must** be specified.

**TET3RT\_TRACE\_MSS** Control the generation of trace messages by MSS functions in the Test Manager. Whether or not trace messages are actually generated depends on the MSS implementation. Typically this variable is set to a value between 0 and 10. The use of this variable is optional; if it is not specified, its value defaults to zero (no trace messages generated).

**TET3RT\_MSS\_RTSYS\_TRACE** Controls the generation of trace messages by Manufacturer-specific subsystems (MSS) functions on the realtime system. Whether or not trace messages are actually generated depends on the MSS implementation. Typically this variable is set to a value between 0 and 10. The use of this variable is optional; if it is not specified, its value defaults to zero (no trace messages generated).

**TET3RT\_MSS\_TRACE2JNL** Specifies whether or not MSS trace message generated by the Test Manager should be printed in the journal as well as appearing on the standard error stream. Whether or not trace message are actually generated depends on the MSS implementation. The possible values are `True` and `False`. The use of this variable is optional; if not specified; its value defaults to `False`.

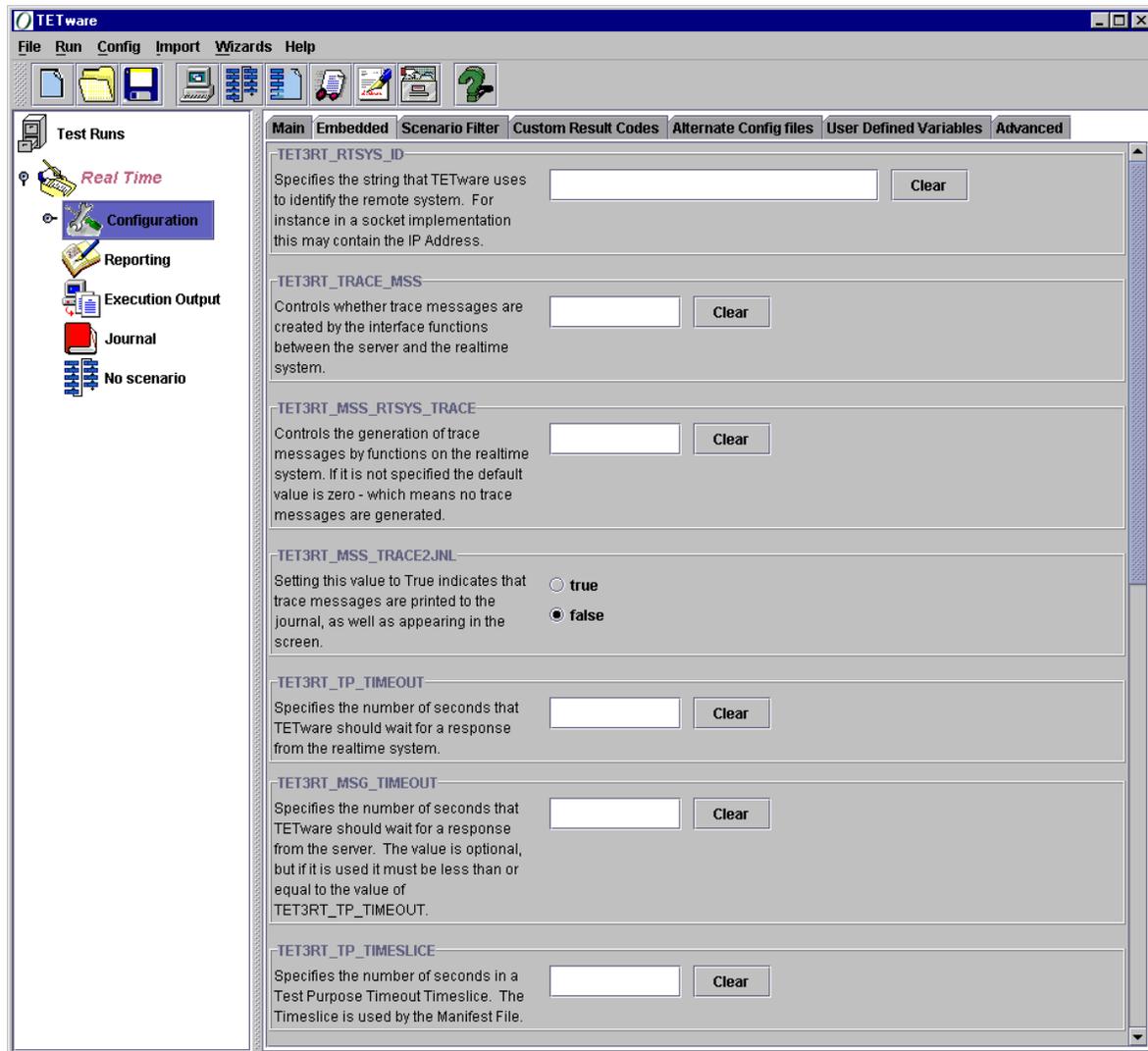


Figure 5: Embedded tab

**TET3RT\_TP\_TIMEOUT** Specifies the number of seconds to be used as the default test purpose timeout. The actual timeout that the Test Manager applies to a test purpose depends on information in the test manifest as well as on the value of this variable; see Section 6.1.6 "Timeout Processing" in the TETware RT, Realtime and Embedded Systems Extension Guide for further details. This variable **must** be specified.

**TET3RT\_MSG\_TIMEOUT** Specifies the number of seconds that TETware should wait for the TCM to respond to a request when in server mode (that is: when the TCM is waiting for the next instruction from the Test Manager). If specified, the value of this variable should be greater than 1

and less than or equal to the value of TET3RT\_TP\_TIMEOUT. The use of this variable is optional; if it is not specified, its value defaults to that specified by TET3RT\_TP\_TIMEOUT.

**TET3RT\_TP\_TIMESLICE** Specifies the number of seconds in a timeslice that is used by the Test Manager when calculating the timeout to be applied to a test purpose. The number of timeslice to be applied to a test purpose may be specified in the test manifest. See Section 6.1.6 "Timeout Processing" in the TETware RT, Realtime and Embedded Systems Extension Guide for further details. This variable **must** be specified.

### 7.1.4 Scenario Filter Tab

The Scenario filter tab allows you to specify Test Cases that you want to include and exclude from the Test Run.

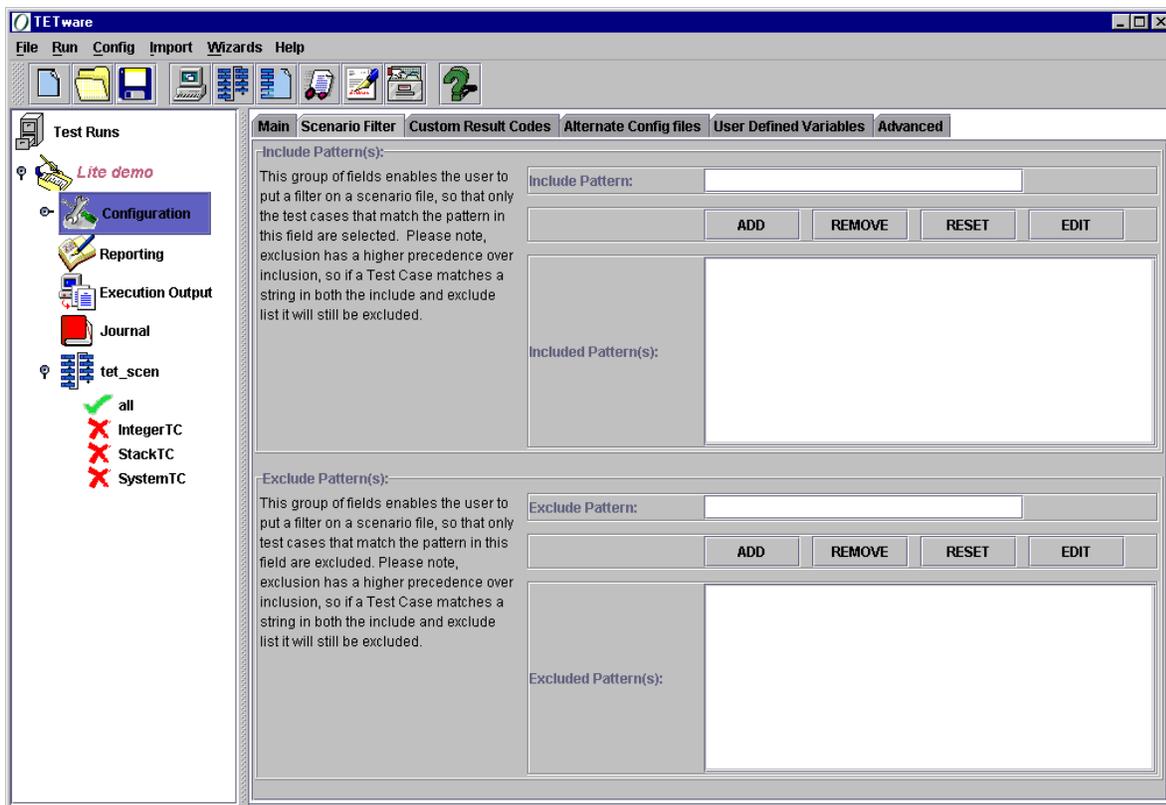


Figure 6: Scenario Filter

**Note:** Exclusion has a higher precedence over inclusion, so if a Test Case matches a string in both the include and exclude lists it will be excluded.

You can specify any number of Test Cases to include or exclude from the Test Run. The Test Cases should be a string of characters that are either a regular expression or a sub-string of a Test Case name.

**Include pattern(s):** This group of fields enables the user to put a filter on a scenario file, so that only the test cases that match the pattern in this field are selected during the Test Run.

**Exclude pattern(s):** This group of fields enables the user to put a filter on a scenario file, so that only test cases that match the pattern in this field are excluded or skipped during Test Run execution.

## 7.1.5 Custom Result Codes Tab

When TETware needs to interpret a test purpose result, it does so by referring to a table of results codes. Initially, this table contains a list of code values, which have standard meanings that may (optionally) be augmented by lists contained in files provided by the test developer.

The result code value is a positive number between 0 and 127. TETware reserves values between 0 and 31 for its own use. Of these Values 0 to 7 are currently used. These standard results codes, all have a `Continue` action indicator.

The Standard Result codes are the following:

0 =	PASS	4 =	UNSUPPORTED
1 =	FAIL	5 =	UNTESTED
2 =	UNRESOLVED	6 =	UNINITIATED
3 =	NOTINUSE	7 =	NORESULT

The Custom Results Codes tab allows you to specify Test Case Result Codes that are not part of the standard Result Codes that TETware uses as it's defaults.

Custom Result Codes should have a value between 32 and 127. You can also give the Result Code some associated text. The value of the Result Code should be placed into the `Code Number` field and the Text relating to it in the `Associated Text` field. An action is also associated with the

Result Code this is indicated by the 'Action' radio buttons; the options are Continue or Abort.

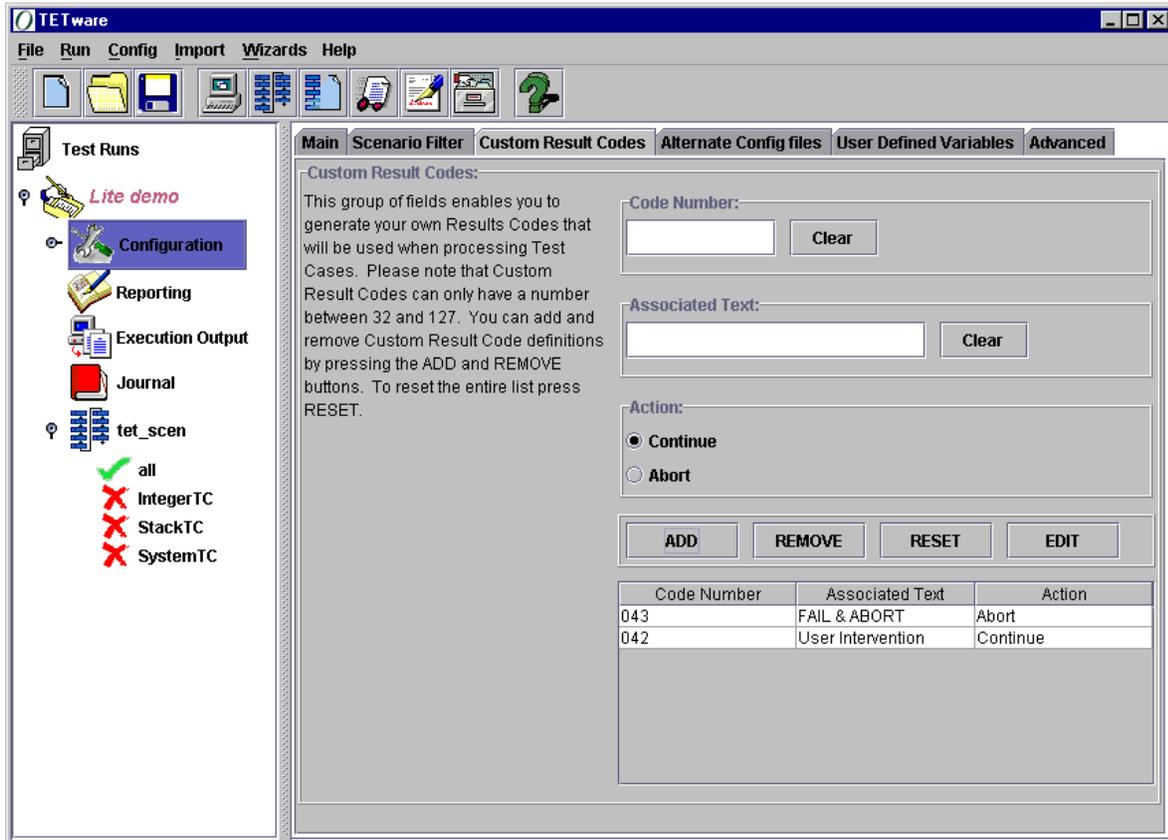
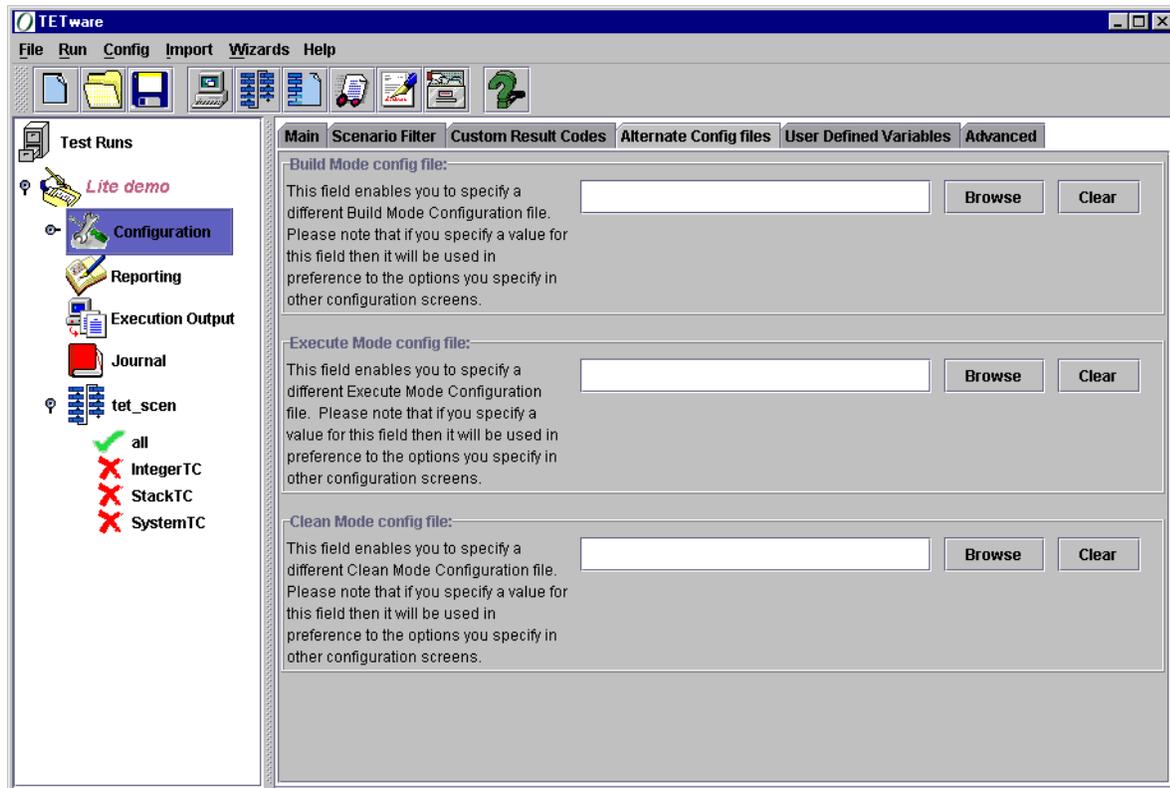


Figure 7: Custom Result codes

## 7.1.6 Alternate Config Files Tab

You are able to specify alternative configuration files to use in place of the ones created by the TETware GUI when a Test Run is executed. These configuration files will be used and no configuration files will be generated by the TETware GUI. You can choose to use one or more alternative configuration files. If you do not specify a configuration file for a mode of operation then the TETware GUI will generate the file for you.



**Figure 8: Alternate Config Files**

The following alternative configuration files can be specified:

- |                |  |
|----------------|--|
| <b>Build</b>   | Specifying a valid file for the Build mode will force the TETware GUI to use this file rather than building a configuration file for the Build mode; |
| <b>Execute</b> | Specifying a valid file for the Execute mode will force the TETware GUI to use this file rather than building an Execute mode configuration file;    |
| <b>Clean</b>   | Specifying a valid file for the Clean mode will force the TETware GUI to use this file rather than building a Clean mode configuration file.         |

### 7.1.7 User Defined Variables Tab

The User Defined Variables tab allows you to specify variables that will be used by your test code in carrying out the Test Run.

Each variable has a name and a value associated with it. In addition to this, a description of that value can be specified. Each variable is applicable to one or more Modes of Operation.

The User Defined Variables tab is shown below in Figure 9:

To add a new variable you need to specify:

Variable Name	Required
Variable Value	Required
Variable Description	Optional
Mode of Operation the variable is applicable to	Required

Once you are satisfied with the variable click ADD to add it to the list of variables. The variable will not be added to the table unless you have specified all the required values.

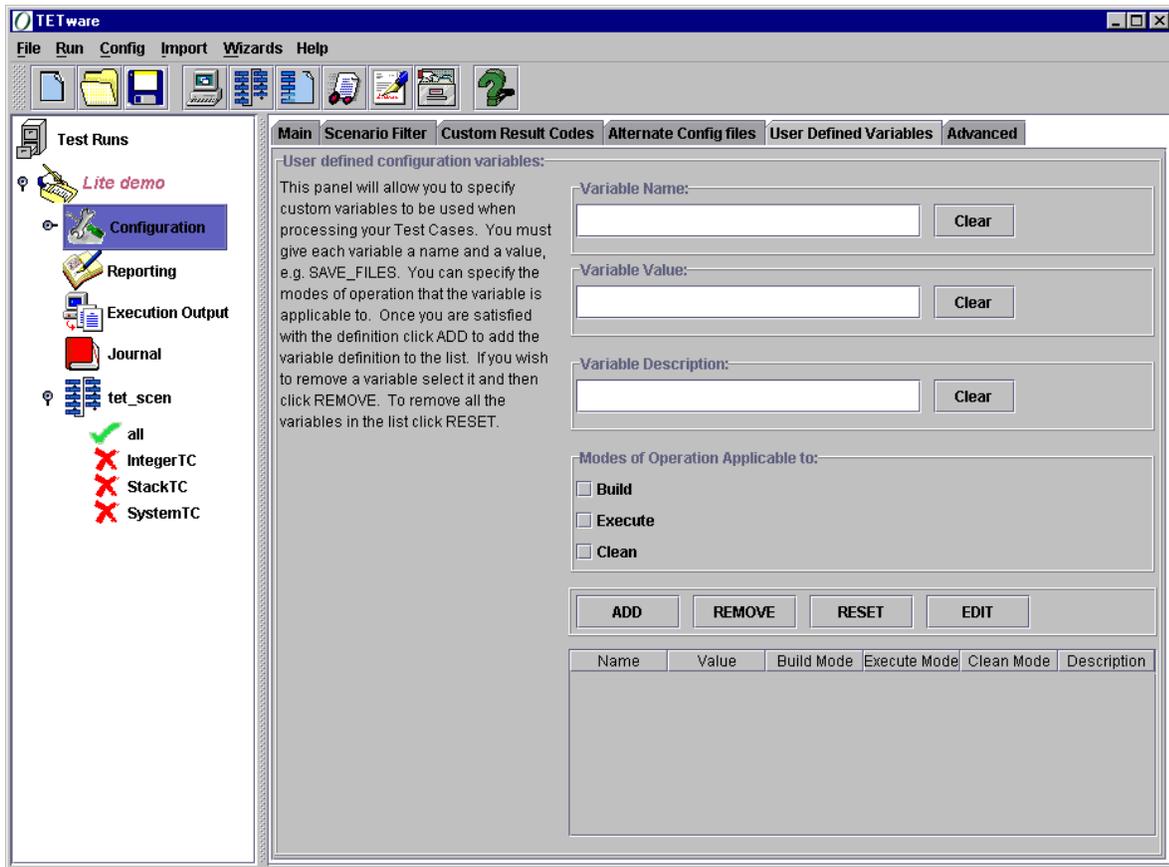


Figure 9: User Defined Variables

## 7.1.8 Advanced Tab

The Advanced tab is the place where all the values considered to be outside the normal scope of operation for the TETware GUI are placed. It may be that certain Users will use these features regularly however it is felt that overall these functions are 'Advanced'.

The Advanced tab contains two groups of items namely:

- Timeout
- User defined Journal file name

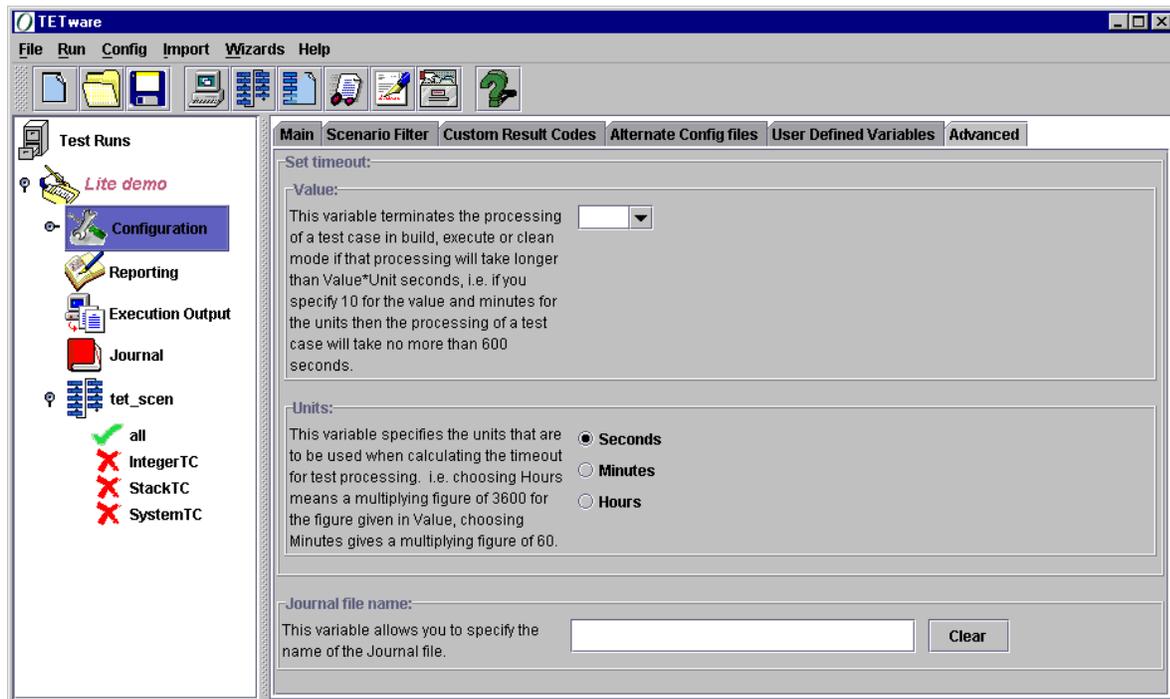


Figure 10: Advanced tab

### 7.1.8.1 Timeout

You are able to specify a timeout that should be passed to TETware to indicate the amount of time that TETware should wait for a Test to complete.

**Value** You can specify a value for the Timeout. The Timeout fields are setup such that the you can specify your own Timeout value or you can select from a pre-defined value given in the drop down box, either 10, 20, 50, 100, 200. If you wish to specify Timeout, it can be typed directly into the first field of the drop down box. The value entered should be a maximum of 5 digits. If the value is invalid, or larger than 5 digits, an error message will be generated when you try to execute the test run, and the value will be ignored.

**Units** You are also able to specify a 'Units' value that will act as a multiplier for the Timeout value. The multipliers are Seconds, Minutes and Hours. Only one multiplier can be specified. Therefore, if you choose to have a timeout of 15 hours, you should type 15 into the value box and choose Hours as the Units value. When you execute the Test Run, the TETware GUI will determine the value of the Timeout in seconds (i.e.  $15 \times 60 \times 60 = 54,000$  Seconds).

### 7.1.8.2 Journal file name

This field allows the User to specify the name of the Journal file that TETware should use for Journal file entries. Please note that you must specify the path, the file name and its extension.

## 7.2 Local System Configuration

Configuration variables may be specified on the Local System. The configuration setting on the Local System is divided into four tabs:

- Local: Environment
- Local: Build Mode
- Local: Execute Mode
- Local: Clean Mode

## 7.2.1 Local: Environment Tab<sup>1</sup>

This, for the Local System, is the place to setup all the environment variables that the TETware GUI will and may need to execute the test run correctly.

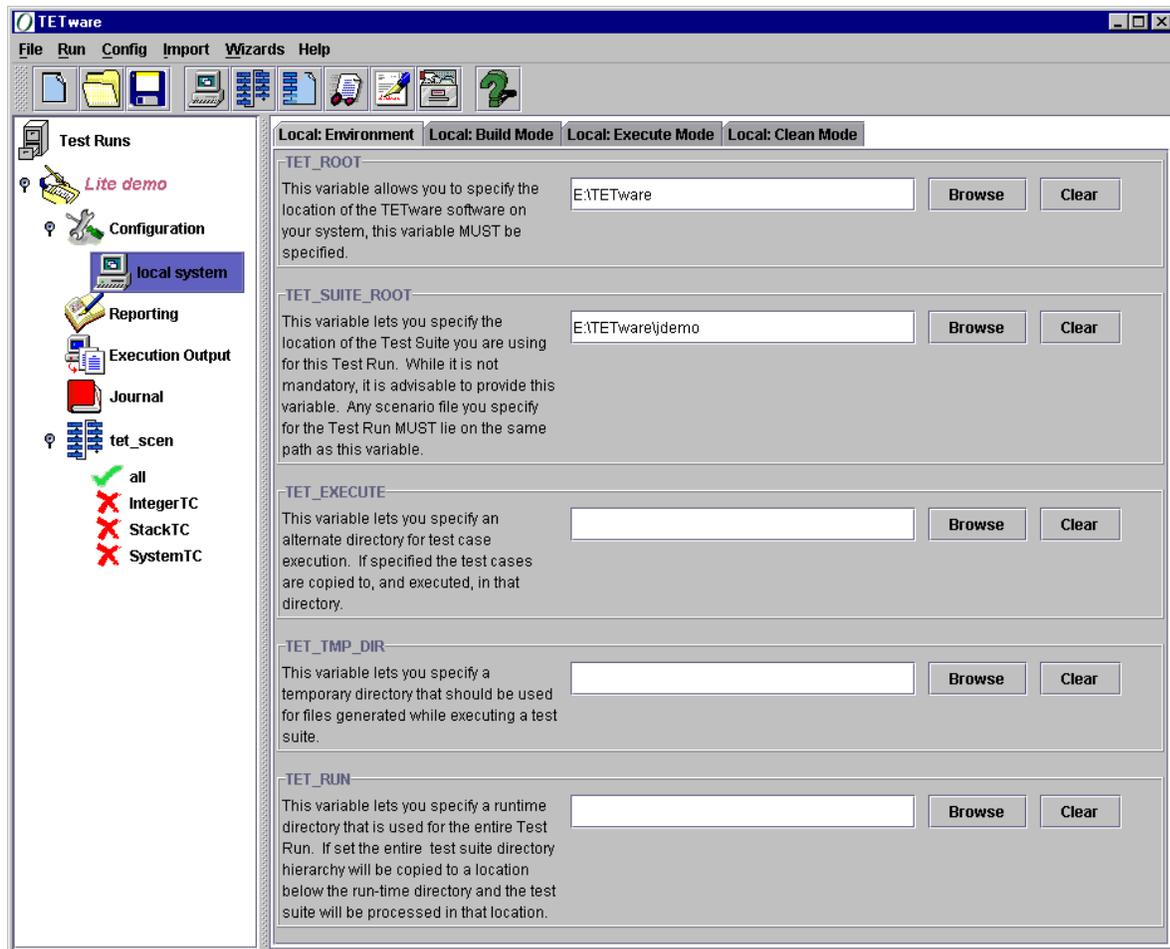


Figure 11: Local: Environment tab

<sup>1</sup> Please note: all variables specified in this tab should be directories, when the Test Run is executed a check is made to ensure that all variables specified in this tab are directories, an error will be displayed if one or more variables are NOT directories and the Test Run execution halted.

All the variables specified in this tab will be passed to the TETware GUI as Environment variables. That is, they will exist as part of the TETware GUI's run-time environment only for the duration that the TETware GUI is running. Any other Test Runs that are executing can and will have their own environment specified.

The following variables can be specified in the Local: Environment tab:

**TET\_ROOT** This variable allows the user to specify to TETware the location of the software, library files and certain configuration files on the users system. This variable **MUST** always be specified for the Local System.

**TET\_SUITE\_ROOT** This variable allows the user to specify to TETware the location of the Test Suite that is to be used for the Test Run. In general, **TET\_SUITE\_ROOT** would sit below **TET\_ROOT** in the directory structure but this is not always true. The Scenario file added to the Test Run must lie somewhere in the directory structure below **TET\_SUITE\_ROOT**. This variable is optional but necessary if the Test Suite does not lie below **TET\_ROOT** in the file system directory structure.

**TET\_EXECUTE** This variable lets you specify an alternate directory for test case execution. If specified the test cases are copied to, and executed in, that directory.

**TET\_TMP\_DIR** This variable lets you specify a temporary directory that should be used for files generated while executing a test suite.

**TET\_RUN** If specified this variable indicates to TETware that it should copy the entire Test Suite to the directory specified and process the Test Suite in that directory.

## 7.2.2 Local: Build Mode

This tab allows you to specify variables that will be used by TETware in Build mode.

The following variables can be specified in Local: Build Mode:

**TET\_BUILD\_TOOL** This variable enables you to specify a tool that TETware should use to build (i.e. compile) test cases, for example make.

**TET\_BUILD\_FILE** This variable allows you to specify additional command line arguments that can be passed to the build tool specified by **TET\_BUILD\_TOOL**. Please note: no validation is performed by the TETware GUI for the relationship between **TET\_BUILD\_TOOL** and **TET\_BUILD\_FILE**; it is the Users responsibility to ensure that they interact correctly.



Figure 12: Local: Build Mode

**TET\_OUTPUT\_CAPTURE** These radio buttons indicate to TETware that the standard output and error streams from the Build tool should be captured and placed in the Journal file. The default value of this variable is `True`. Setting this variable to `False` tells TETware not to record test case output in the journal file.

**TET\_API\_COMPLIANT** These radio buttons allow you to specify to TETware whether the Build tool and Test Cases are to be built to conform to the TETware API, i.e. they use the TETware API to output information to the Journal file. The default value of this variable is `False`. Setting the value to `False` indicates that all Test Cases and relevant tools used in the Test Run conform to a specific TETware API. Those tools and Test Cases are then expected to use the API to generate diagnostics and register results. If set to `True` the printing of messages to the journal file and the generation of the test purpose result will be based on the exit status of the Test Case (where `PASS = 0`, and `FAIL = non 0`).

**TET\_PASS\_TC\_NAME** These radio buttons allow you to specify to TETware whether the name of the current Test Case should be passed to the Build tool specified. The default value of this variable is `False`. Setting this value to `True` indicates that the name of the Test Case to be processed should be passed when executing a Build or Clean tool.

**TET\_EXPAND\_CONF\_VARS** The value of this variable determines whether or not `tcc` performs configuration variable expansion on variables defined in the per-mode configurations. If `true`, `tcc` replaces the string `${variable-name}` in a configuration variable assignment with the value of the variable `variable-name`. If `false`, `tcc` does not treat the string `${variable-name}` in a configuration variable assignment specially. The default value is `false`. For further information on this feature is presented in Section 5.8 of the TETware Programmers Guide.

## 7.2.3 Local: Execute Mode

This tab allows you to specify variables that will be used by the TETware in Execute mode.

**TET\_EXEC\_TOOL** This variable enables you to specify a tool that TETware should use to execute test cases.

**TET\_EXEC\_FILE** This variable allows you to specify additional command line arguments that can be passed to the execute tool specified by `TET_EXEC_TOOL`. Please note: no validation is performed by the TETware GUI for the relationship between `TET_EXEC_TOOL` and `TET_EXEC_FILE`; it is the Users responsibility to ensure that they interact correctly.

**TET\_OUTPUT\_CAPTURE** This radio button variable indicates to TETware whether the standard output and error streams from the Execute tool should be captured and placed in the Journal file. The default value of this variable is `True`. Setting this value to `False` indicates that the standard output and the standard error output from Test Cases should not be captured and placed in the journal file.

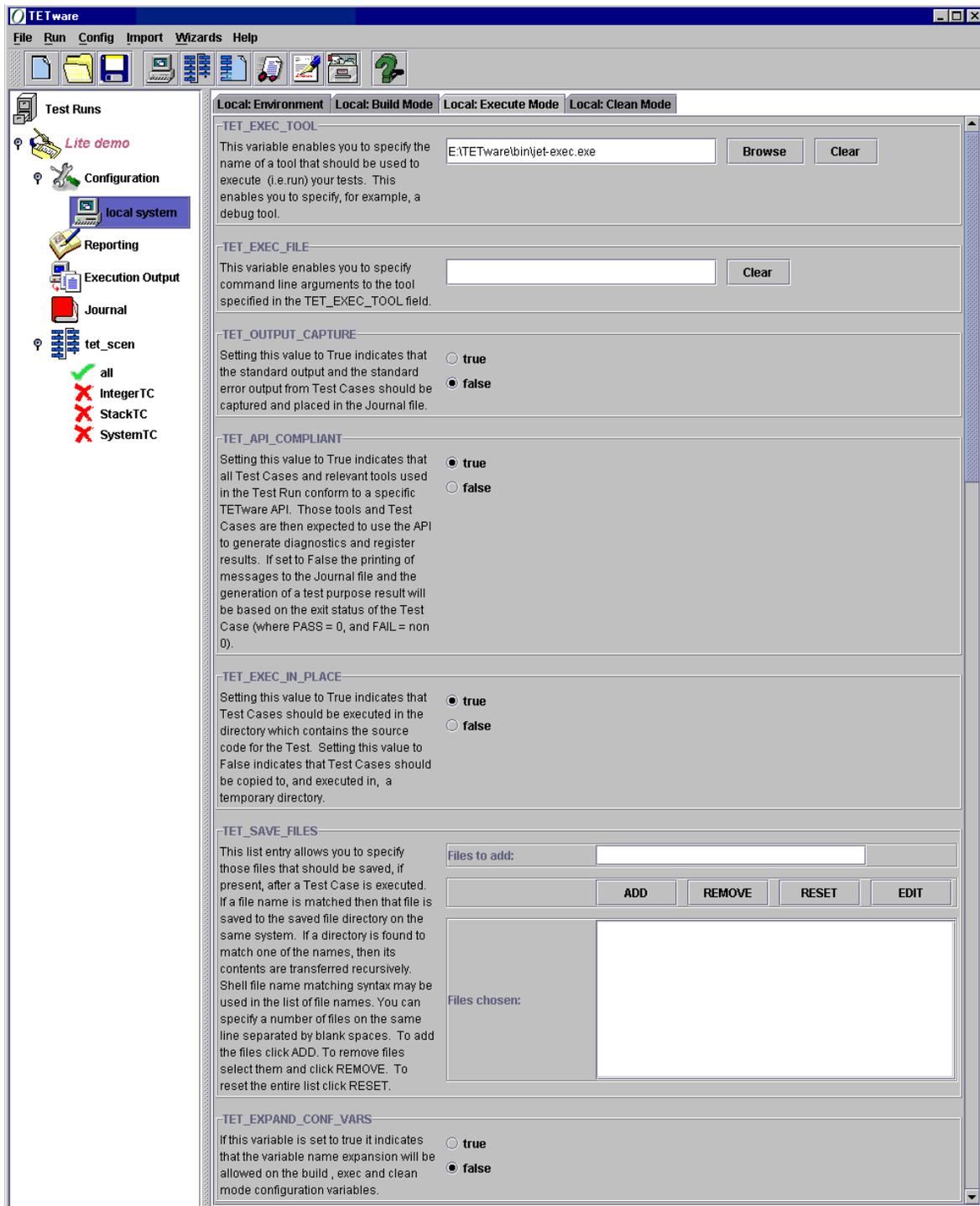


Figure 13: Local: Execute Mode

**TET\_API\_COMPLIANT** These radio buttons allow you to specify to TETware whether the Execute tool and Test Cases to be built conform to the TETware API, i.e. they use the TETware API to output information to the Journal file. The default value of this variable is `False`. Setting this value to `True` indicates that all Test Cases and relevant tools used in the Test Run conform to a specific the TETware API. Those tools and Test Cases are then expected to use the API to generate diagnostics and register results. If set to `False` the printing of the journal file and the generation of a Test purpose result will be based on the exit status of the Test Case (where `PASS = 0`, and `FAIL = non 0`).

**TET\_EXEC\_IN\_PLACE** These radio buttons indicate to TETware that it should execute Test Cases in the directory that contains the source code for the Tests. Setting this value to `True` indicates that Test cases should be executed in the directory, which contains the source code for the Test. Setting the value to `False` indicates that the Test Case should be copied to, and executed in, a temporary directory.

**TET\_SAVE\_FILES** You can specify a number of files that TETware should save after Test execution (if it is able to find those files). You can specify directories or normal shell matching syntax (i.e. `*`) as part of the list.

**TET\_EXPAND\_CONF\_VARS** The value of this variable determines whether or not `tcc` performs configuration variable expansion on variables defined in the per-mode configurations. If `true`, `tcc` replaces the string `${variable-name}` in a configuration variable assignment with the value of the variable `variable-name`. If `false`, `tcc` does not treat the string `${variable-name}` in a configuration variable assignment specially. The default value is `false`. For further information on this feature is presented in Section 5.8 of the TETware Programmers Guide.

## 7.2.4 Local: Clean Mode

This tab allows you to specify variables that will be used by TETware in Clean mode.

The following variables can be specified in the Local: Clean Mode tab:

**TET\_CLEAN\_TOOL** This variable enables you to specify a tool that TETware should use to clean test cases.

**TET\_CLEAN\_FILE** This variable allows you to specify additional command line arguments that can be passed to the clean tool specified by `TET_CLEAN_TOOL`. Please note: no validation could

be performed by the TETware GUI for the relationship between `TET_CLEAN_TOOL` and `TET_CLEAN_FILE`; it is the Users responsibility to ensure that they interact correctly.

**TET\_OUTPUT\_CAPTURE** These radio buttons indicate to TETware whether the standard output and error streams from the Clean tool should be captured and placed in the Journal file. The default value of this variable is `True`. Setting this value to `False` indicates that the standard output and the standard error output from Test Cases should not be captured and placed in the journal file.

**TET\_API\_COMPLIANT** These radio buttons allow you to specify to TETware whether the Clean tool and Test Cases to be built conform to the TETware API, i.e. they use the TETware API to output information to the Journal file. The default value of this variable is `False`. Setting the value to `True` indicates that all Test Cases and relevant tools used in the Test Run conform to a specific TETware API. Those tools and Test Cases are then expected to use the API to generate diagnostics and register results. If set to `False` the printing of messages to the journal file and the generation of the test purpose result will be based on the exit status of the Test Case (where `PASS = 0`, and `FAIL = non 0`).

**TET\_PASS\_TC\_NAME** These radio buttons allow you to specify to TETware whether the name of the current Test Case should be passed to the Clean tool specified. The default value of this variable is `FALSE`. Setting this value to `TRUE` indicates that the name of a Test Case to be processed should be passed when executing a Build or Clean tool.

**TET\_EXPAND\_CONF\_VARS** The value of this variable determines whether or not `tcc` performs configuration variable expansion on variables defined in the per-mode configurations. If `true`, `tcc` replaces the string `${variable-name}` in a configuration variable assignment with the value of the variable `variable-name`. If `false`, `tcc` does not treat the string `${variable-name}` in a configuration variable assignment specially. The default value is `false`. For further information on this feature is presented in Section 5.8 of the TETware Programmers Guide.

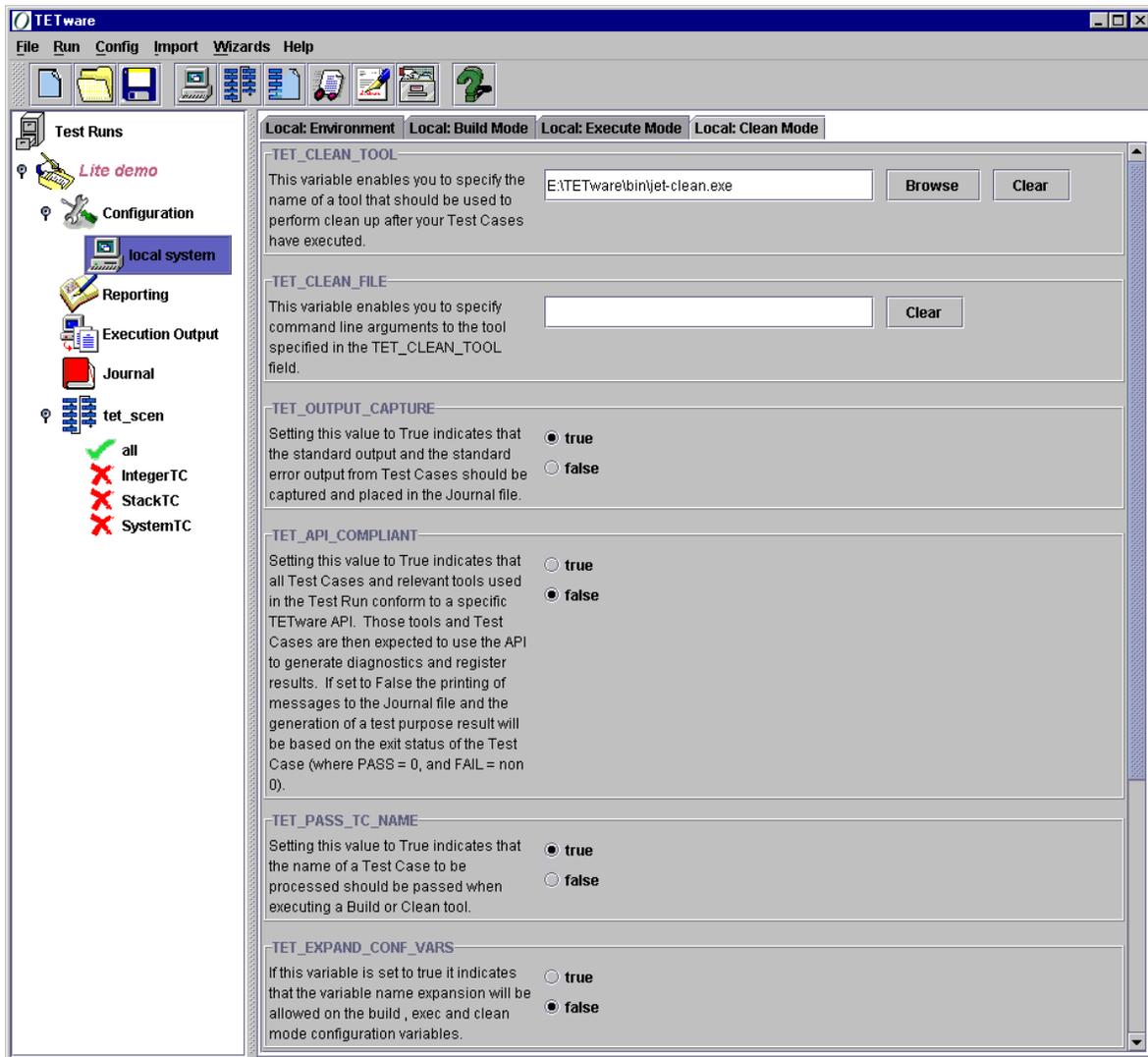


Figure 14: Local: Clean Mode.

### 7.3 Importing Configuration files

For existing TETware and TET Users an import function has been provided for importing configuration values. Importing configuration is performed on a per Test Run basis. The following types of configuration file are supported for import:

- Build mode
- Execute mode
- Clean mode
- Distributed config (Distributed TETware only)

For each configuration file imported the file is parsed and values that the TETware GUI understands are placed into the configuration for the given Test Run. Values that the TETware GUI does not understand are added to the User defined variables for a Test Run. All lines in the configuration file that start with # are ignored as comment lines. Comment lines are not imported since it is impossible to define rules that would capture the many different ways of commenting that are possible.

### 7.3.1 User Defined Variables

Imported User Defined Variables will be added to the table in the User Defined Variables tab, for more information on User Defined Variables please see Section 7.1.7. The mode of operation that is given to the imported User Defined Variable is determined by the type of configuration file that is imported. For instance, if a Build mode configuration file is imported then all User Defined Variables in that file will be given a mode of Build in the User Defined Variables table.

### 7.3.2 Distributed config files

It is also possible to import a distributed configuration file (usually called `tetdist.cfg`). When importing a distributed configuration file you **MUST** have all the remote systems that have values in the distributed configuration file already added to the Test Run. i.e. if the distributed configuration file contains a value `TET_REM001_TET_ROOT` then a remote system with a system number of 001 must already be present in the Test Run for that value to be imported correctly. (See also Section 9.3.4).

### 7.3.3 How to import a configuration file

To import a configuration file you can either:

- Use the Import menu on the menu bar, from there you can choose either Build, Execute, Clean or Distributed configuration files to import.

- Right-click on the Configuration node in the Test Run. A pop-up menu containing items for Build, Execute, Clean and Distributed configuration files are shown.

Regardless of the method of import, a file selection dialog is then displayed from which the User can browse the file system and choose a configuration file. A file filter of \*.cfg is also given to filter for configuration files that are given the default extension of cfg.

**Please note:** the parser for the configuration files will split any valid lines (lines not blank and not beginning with #) on "=", so any malformed lines will be added without verification. It is the Users responsibility to ensure that the configuration file is correct and contains values that are valid and have meaning to the TETware GUI and their Test Cases.

## 8 Scenario File(s)

To execute a Test Run, a scenario file **MUST** be included.

### 8.1 What is a Scenario File?

For a Test Run to be useful, one or more scenarios must be selected from a Scenario file that has been added into the Test Run. Scenario files are used in manipulating the Test Run.

By default there is a scenario called `all`. This `all` scenario has a special meaning to TETware, if it is defined in the scenario file then it will be mutually exclusive with the other scenarios in the scenario file. This is to prevent locking problems and potential problems with scenario looping. If the `all` scenario is chosen then all other scenarios will be set to not selected and the `all` scenario - only - will be selected. If another Scenario is selected while the `all` scenario is selected, the `all` Scenario will be set to not selected and the chosen scenario will be selected.

It is possible to select all of the scenarios in a Scenario file by right clicking on the Scenario node in the tree and clicking `Include all scenarios`, this will set all of the scenarios in the Scenario file as selected.

### 8.2 Adding a Scenario File

To add a scenario file the User can:

- Use the icon on the tool bar .
- Use the `Add Scenario File` menu item in the `Config` menu.
- Use the Keyboard Shortcut `Alt-L`
- Right-click on the Test Run tree node and click on the `Add Scenario File` menu item.<sup>2</sup>

---

<sup>2</sup> Please note that these menu items will only be available if there is **NOT** a scenario file already added to the Test Run. If a scenario file has already been included in the Test Run, it must be deleted before an alternative scenario file can be added.

Whichever method is chosen, a file selection dialog box will be displayed for you to choose the Scenario File needed.

There are two file filters provided to allow the User to find and select scenario files quickly, they are:

<code>scen.*</code>	This will filter for all files STARTING with <code>scen.</code>
<code>*.scn</code>	This will filter for all files ENDING with <code>.scn</code> .

Once a scenario file has been selected by clicking `Open`, a check will be made to ensure that the chosen file is not a directory and that it exists. If those criteria are NOT met then a dialog box will be displayed with a message ‘Scenario file is not valid – Error [*File name*] is not a valid file’.

If a file is chosen and it is valid, (i.e. it exists and is a file not a Directory) then it will be parsed by the Scenario file parser. The name of the Scenario file will be added to the Scenario file node and all the scenarios found added to that Node as children<sup>3</sup>. When a scenario file is added, the top scenario (usually “all”) is selected (for details of the “all” scenario please see Section 8.1). Note that only one scenario file may be included in a Test Run, but a scenario file can include multiple scenarios.

### 8.3 Selecting Test Cases from a Scenario File

To select Test Cases from within a Scenario file, you will need to

Expand the Scenario Node

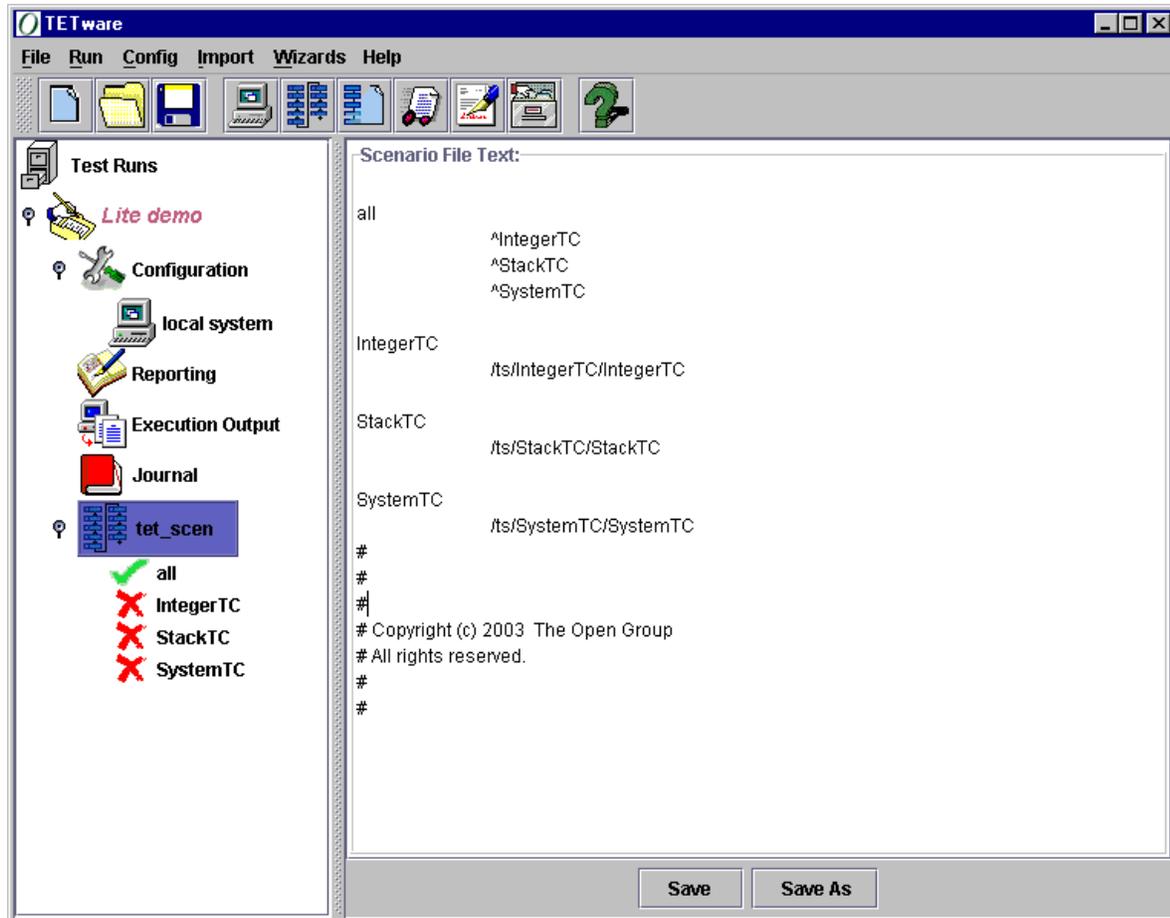
- Place your Cursor on the Test Case to select.
- Click on the test case.
- The selected case will then change from a  to a .

---

<sup>3</sup> Please note: once a Scenario file has been added to the Test Run, the menu items for adding a Scenario file will be disabled until that Scenario file is removed.

## 8.4 Editing a Scenario File

To Edit a Scenario file, you will need to expand the Scenario Node.



**Figure 15: Editing Scenario File**

- Click on a selected Scenario or on any of the expanded Test Cases
- Place your cursor on the Test Case field
- Start editing by typing any changes you wish to make.
- Then click on **SAVE** to save using the existing scenario file name or click on the **SAVE AS** button to save using another name.

If `SAVE` is selected a message box confirms the name of the saved Scenario file. If `SAVE AS` is selected a file chooser box appears giving the user an opportunity to save the edited Scenario file using another name.

## 8.5 De-Selecting Test Cases from the Scenario File

To De-select Test Cases from within a Scenario file, you will need to:

- Expand the Scenario Node
- Place your Cursor on the Test Case to select
- Click on the Test Case
- The selected case should then change from a  to a  to show that it has been deselected.

## 8.6 Creating a Scenario File

A scenario file can be created automatically by using a file parser. The generated scenario file is in a simple format, without directives. It can be edited and saved, including saved as a different file name if the user wishes.

To Create a Scenario File you can:

- Use the icon on the menu bar .
- Use the `Create Scenario File` menu item in the `Config` menu.
- Right-click on the `Test Run` tree node at the `No Scenario` point and click on the `Create Scenario File` menu item.
- Use the Keyboard Shortcut `Alt-K`

(Please note that these items will only be available if there is NOT a scenario file already added to the `Test Run`)

Whichever method is chosen, the create Scenario file tab is displayed.

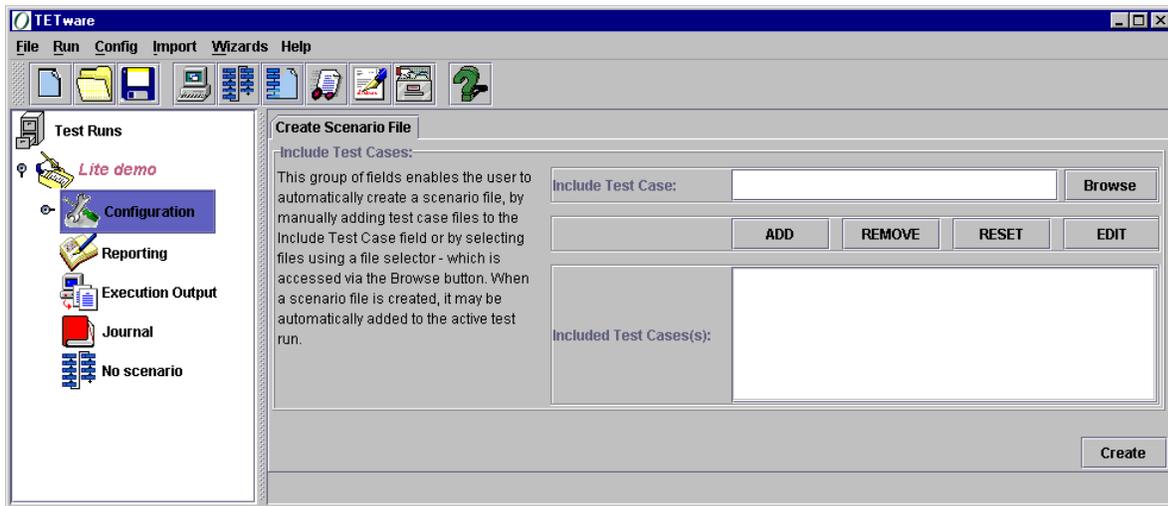


Figure 16: Create Scenario file Tab

### 8.6.1 Including Test Cases in the Scenario File Creation (ADD)

There are two main ways of creating a Scenario file on this screen, either

- Type in the path and name of the Test Case(s) or
- Browse and select Test Case(s).

(Please note that, you need to have specified where TET\_ROOT is, under Local System: Environment TET\_ROOT settings, see Section 7.2.1).

- If you choose to type into the field `Include Test Case(s):`, you will need to click on the ADD button and the Test Case will be included in the scenario file being created.
- If you choose to browse, then the Select Test(s) dialog box opens, which will allow you to choose test cases by putting a check mark in the appropriate check box.

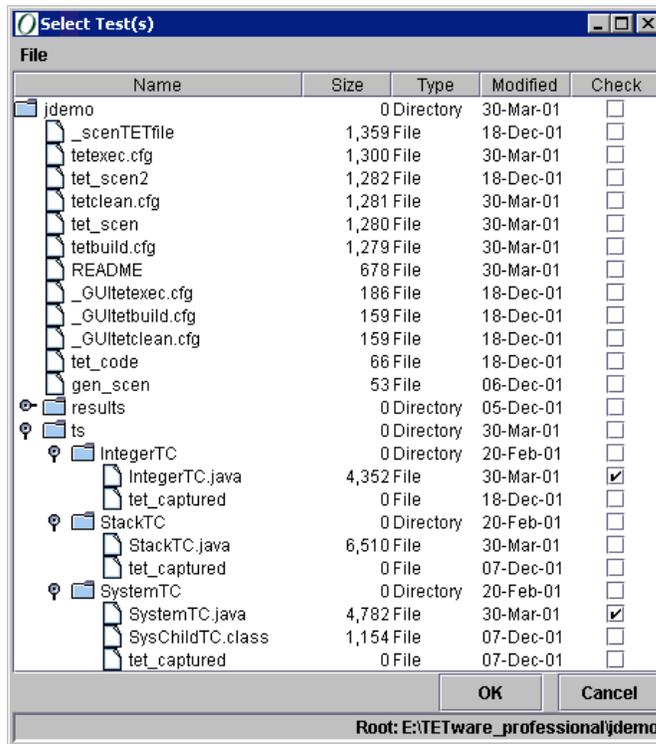


Figure 17: Scenario File Chooser

- You should then click on OK and the file(s) will be transferred to the Included Test Case(s) field. Alternatively, if you click on Cancel, the Select Test(s) box is closed and you are returned to the Create Scenario File tab.

## 8.6.2 Removing Test Cases from Scenario File Creation (REMOVE)

To remove any of the previously added Test cases in the Scenario file, you will need to select them from the Included Test Case(s) field and click REMOVE, the files will then be removed.

To remove multiple files you can hold the 'Shift' or 'Ctrl' key and select rows in the table using your mouse or the arrow key, if you hold 'Shift' then you can select a contiguous list of variables but if you hold 'Ctrl', you can choose a non-contiguous list of files. To remove the selected files, press REMOVE.

### 8.6.3 Resetting the Whole File Selection Process (RESET)

If you want to reset the whole file selection, all you need to do is to press on the `RESET` button and all the included test case(s) will disappear leaving the field empty.

### 8.6.4 Editing the Selected Variable (EDIT)

If you wish to change some of the parameters in your selection, you should highlight the file to be edited, and click on the `EDIT` button. This will cause the file to appear in the field `Include Test Case` field, it can then be edited in this field.

After editing, you will again need to press on the `ADD` button to include the edited file in the scenario that is being created.

### 8.6.5 Saving the Scenario File (CREATE)

After you have browsed, edited, and added the selected test cases into the scenario file to be created. You need to click on the `Create` button to generate and save the Scenario file.

When the `Create` button is pressed, a `Save As` dialog box will appear to provide the option to save the scenario file into any drive and directory. You should save it in either your `TET_ROOT` or `TET_SUITE_ROOT` directory.

After making your selection and naming the scenario file created, a confirmation dialog box `Save Scenario File` will be presented. Click on `OK` to close it.

A dialog box is then presented providing the option of adding the new Scenario file to the Test Run tree. If you select `Yes`, the Scenario file is added to the tree and the generated Scenario, all is selected. Selecting `No` closes the dialog box and returns you to the `Create Scenario File` tab.

Once added the Scenario file may be edited in the Scenario File Text screen (see Section 8.4) and can be saved using the same name, or saved as an alternative file name.

## 8.7 Removing a Scenario File

There are three ways to remove a Scenario file:

- Use the `Remove Scenario File` menu item in the `Config` menu.
- Right-click on the scenario node in the tree and click the `Remove Scenario File` menu item.
- Press the `Delete` key when the Scenario file is highlighted.

Whichever method is used, a dialog box will be displayed for you to confirm removal of the Scenario file.

If the `Yes` button is clicked then the Scenario file will be removed from the Test Run. All the child scenario nodes will also be removed. The label on the Scenario file node in the Test Run tree will be set to `No Scenario`. If `No` is clicked then the Scenario file will not be removed from the Test Run. The Scenario Node settings will remain as they were.

Once the Scenario file has been removed, the menu items relating to the addition of a Scenario file will be re-activated, and the menu items relating to remove a Scenario file will be de-activated.

## 9 Remote Systems (Distributed TETware only)

Configuration variables may be specified on any Remote System that is to participate in remote or distributed testing. In this context, the Remote Systems are other systems on which test cases or test case parts are run other than the Local System. **The Local System always has a system ID of zero (000)**; other system IDs always refer to Remote Systems. In this context, the Local System is defined as the one on which TETware is invoked. Usually, different TETware systems are located on different physical machines. However, there is no reason why a single machine should not have several different system IDs.

Configuration variable assignments made on the Local System are propagated to each of the Remote Systems; however, configuration variable assignments made on a Remote System normally have precedence over those that are propagated from the Local System.

### 9.1 Adding a Remote System

Please note that adding a Remote System is only appropriate if you are using the Distributed TETware GUI.

Once the TETware GUI has been loading and you have already created or selected a Test Run, then you are able to add a Remote System by any of the 4 options listed below:

- Click on the icon to create a Remote System  using the Remote System Wizard
- Use the Keyboard Shortcut Alt-M to access the Remote System Wizard
- Use the Add Remote System menu item in the Config menu at the tool bar
- Right Click on the mouse while on the Configuration Tree Node and select Add Remote System.

If you click on the icon to create a Remote System, you will need to follow the on screen information and set up all the necessary variables that are requested, (some of these fields are mandatory while others are not).

#### 9.1.1 Using the Remote Wizard to Add a Remote System

If you decided to use this method, you will be taken through a series of dialog boxes prompting you to enter configuration information. To do this effectively you will need to refer to Section 9.2 'Configuration Settings on the Remote System'.

## 9.1.2 Using the Tree Node or Menu Bar to Add a Remote System

If using the menu bar then click on `Config`, point and click on `Add Remote System`. Or right click on the mouse while your cursor is over `Configuration` in the tree node and select `Add Remote System`.

1. A dialog box will appear with the message 'Please enter a name for the Remote System'. It also reminds you to note that the Remote System name must be a fully qualified domain name or IP address. After entering the name or IP address, please click on `OK` to continue or `Cancel` to abort.
2. If you click on `OK`, without entering the information required, you will be prompted by an error message 'No Remote System name given or specified'. If details are entered then it will prompt you to enter a Remote System number between 1 and 999.
3. If number 0 is entered here it will give an error message: 'Error: System number 000 is reserved for the Local System' or if a number greater than 999 was entered, it returns an error message 'Remote System number is invalid'. When a valid number is given, a Remote System will be created.

## 9.2 Configuration Settings on the Remote System

Configuration settings on Remote Systems are similar to those on the Local System (Section 7.2). There are 4 tabs and they are:

- Remote: Environment tab
- Remote: Build Mode field tab
- Remote: Execute Mode tab
- Remote: Clean Mode tab

## 9.2.1 Remote: Environment

This tab allows you to specify variables for the operation of TETware on the Remote System concerned (please refer to Figure 18).

**TET\_ROOT** This variable specifies to the TETware the location of library files and certain configuration files that must exist under the TET\_ROOT directory for the Remote System. This variable **MUST** always be specified for a Remote System participating in testing.

**TET\_TSROOT** This variable is only found on Remote Systems and is used to indicate the location of the Test Suite root directory on the Remote System. **It must be specified for each Remote System participating in the Test Run.**

**Mapping** This specifies the mapped path of the TET\_ROOT directory for the Remote System.

**Note:** It may be that the Remote System to be used in testing is not accessible via the file system, i.e. the remote file system is not mounted or no network drive has been mapped, in which case the directory to use should be typed directly into the text field. Use the **Clear** button to clear the contents of the relevant field.

**TET\_SUITE\_ROOT** For Remote Systems this variable is not used by TETware but if specified is passed to Test Cases and tools in the environment as a communication variable.

**TET\_EXECUTE** If specified this variable indicates to TETware that it should perform Test Case execution in the directory specified on the Remote System. This variable is passed as a communication variable.

**TET\_TMP\_DIR** If specified this variable indicates to TETware that it should place all files generated by Test Case activity under the directory specified. This variable is passed as a communication variable.

**TET\_RUN** If specified this variable indicates to TETware that it should copy the entire Test Suite to the directory specified and process the Test Suite in that directory. This variable is passed as a communication variable.

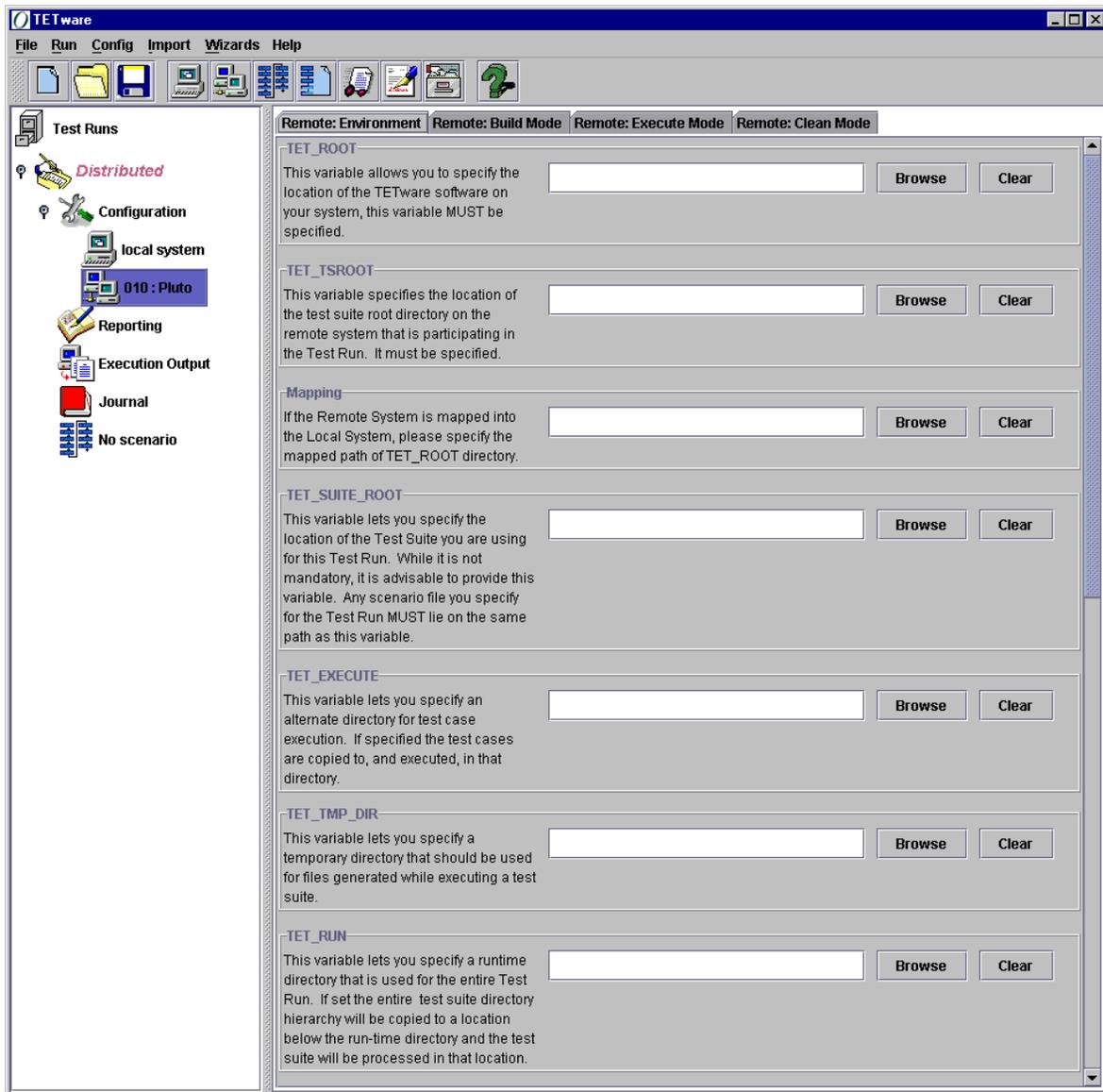


Figure 18: Remote: Environment

## 9.2.2 Remote: Build Mode

This tab allows you to specify variables that will be used by TETware in Build mode for the Remote System specified.

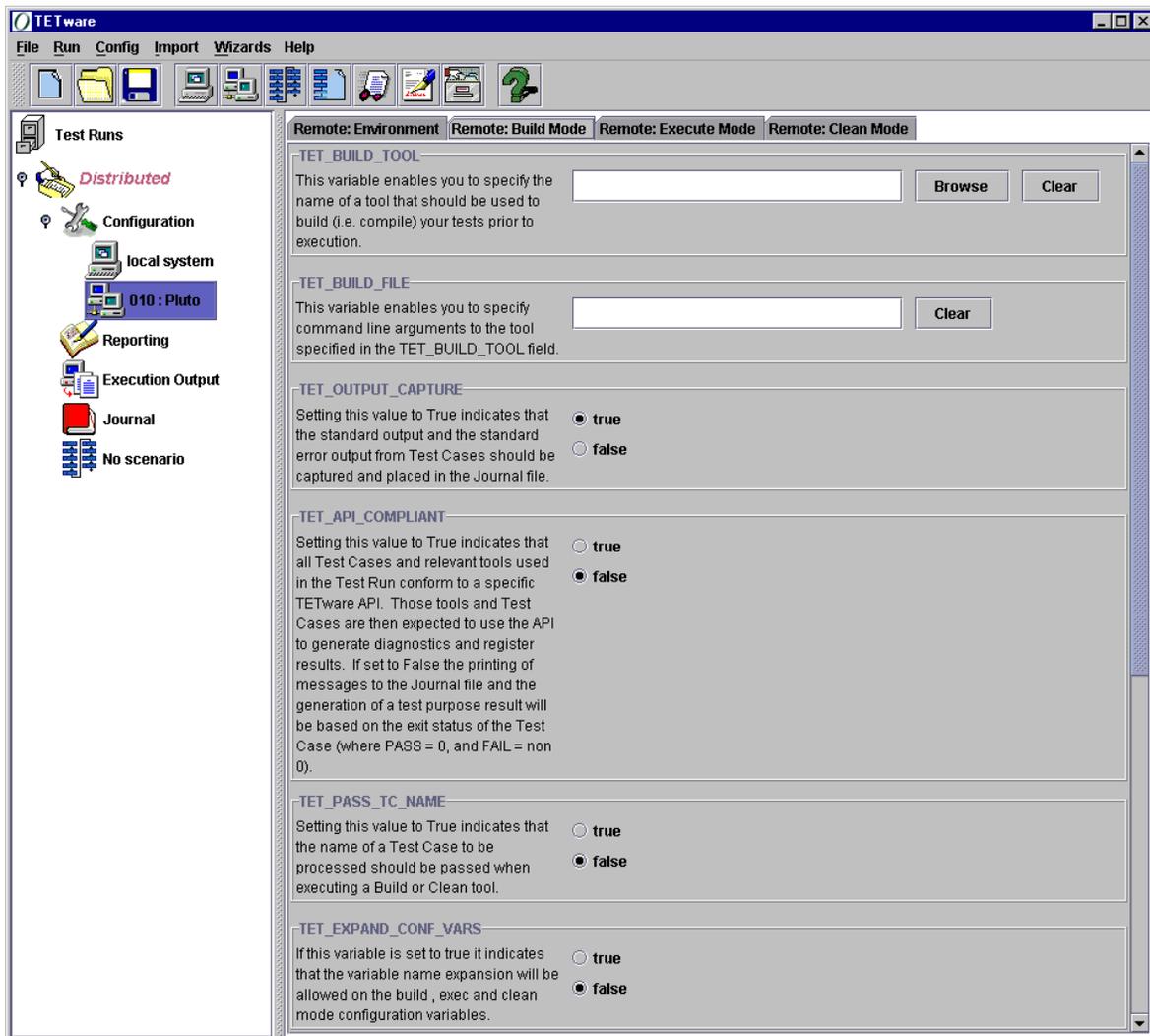


Figure 19: Remote: Build Mode

**TET\_BUILD\_TOOL** This variable enables you to specify a tool that TETware should use to build test cases, for example 'make'.

**TET\_BUILD\_FILE** This variable allows you to specify additional command line arguments that can be passed to the build tool specified by `TET_BUILD_TOOL`. Please note: no validation is performed by the TETware GUI for the relationship between `TET_BUILD_TOOL` and `TET_BUILD_FILE`, it is the Users responsibility to ensure that they interact correctly.

**TET\_OUTPUT\_CAPTURE** These radio buttons indicate to TETware whether the standard output and error streams from the Build tool should be captured and placed in the Journal file. The default value of this variable is `True`. Setting this value to `False` indicates that the standard output from Test Cases should not be captured and placed in the Journal file.

**TET\_API\_COMPLIANT** These radio buttons allow you to specify to TETware whether the Build tool and Test Cases to be built conform to the TETware API, i.e. they use the TETware API to output information to the Journal file. The default value of this variable is `False`. Setting the value to `True` indicates that all Test Cases and relevant tools used in the Test Run conform to a specific TETware API. Those tools and Test Cases are then expected to use the API to generate diagnostics and register results. If set to `False` the printing of messages to the journal file and the generation of the test purpose result will be based on the exit status of the Test Case (where `PASS = 0`, and `FAIL = non 0`).

**TET\_PASS\_TC\_NAME** These radio buttons allow you to specify to TETware whether the name of the current Test Case should be passed to the Build tool specified. The default value of this variable is `False`. Setting this value to `True` indicates that the name of the Test Case to be processed should be passed when executing a Build or Clean tool.

**TET\_EXPAND\_CONF\_VARS** The value of this variable determines whether or not `tcc` performs configuration variable expansion on variables defined in the per-mode configurations. If `true`, `tcc` replaces the string `${variable-name}` in a configuration variable assignment with the value of the variable `variable-name`. If `false`, `tcc` does not treat the string `${variable-name}` in a configuration variable assignment specially. The default value is `false`. For further information on this feature is presented in Section 5.8 of the TETware Programmers Guide.

### 9.2.3 Remote: Execute Mode

This tab allows you to specify variables that will be used by TETware in Execute mode.

**TET\_EXEC\_TOOL** This variable enables you to specify a tool that TETware should use to execute test cases.

**TET\_EXEC\_FILE** This variable allows you to specify additional command line arguments that can be passed to the execute tool specified by **TET\_EXEC\_TOOL**. Please note: no validation is performed by the TETware GUI for the relationship between **TET\_EXEC\_TOOL** and **TET\_EXEC\_FILE**, it is the Users responsibility to ensure that they interact correctly.

**TET\_OUTPUT\_CAPTURE** These radio buttons indicate to TETware whether the standard output and error streams from the Execute tool should be captured and placed in the Journal file. The default value of this variable is **True**. Setting this value to **False** indicates that the standard output and the standard error output from Test Cases should not be captured and placed in the Journal file.

**TET\_API\_COMPLIANT** These radio buttons allow you to specify to TETware whether the Execute tool and Test Cases to be built conform to the TETware API, i.e. they use the TETware API to output information to the Journal file. The default value of this variable is **False**. Setting this value to **True** indicates that all Test Cases and relevant tool used in the Test Run conform to a specific TETware API. Those tools and Test Cases are then expected to use the API to generate diagnostics and register results. If set to **False** the printing of the Journal file and the generation of a Test purpose result will be based on the exit status of the Test Case (where **PASS** = 0, and **FAIL** = non 0).

**TET\_EXEC\_IN\_PLACE** These radio buttons indicate to TETware that it should execute Test Cases in the directory that contains the source code for the Tests. Setting this value to **True** indicates that Test cases should be executed in the directory that contains the source code for the Test. Setting the value to **False** indicates that the Test Case should be copied to and executed to and executed in a temporary directory.



Figure 20: Remote: Execute Mode

**TET\_SAVE\_FILES** You can specify a number of files that TETware should save after Test execution (if it is able to find those files). You can specify directories or normal shell matching syntax (i.e. \*) as part of the list.

**TET\_EXPAND\_CONF\_VARS** The value of this variable determines whether or not `tcc` performs configuration variable expansion on variables defined in the per-mode configurations. If `true`, `tcc` replaces the string `${variable-name}` in a configuration variable assignment with the value of the variable `variable-name`. If `false`, `tcc` does not treat the string `${variable-name}` in a configuration variable assignment specially. The default value is `false`. For further information on this feature is presented in Section 5.8 of the TETware Programmers Guide.

**TET\_TRANSFER\_SAVE\_FILES** This radio button variable indicates to TETware that it should transfer the files specified in the `TET_SAVE_FILES` variable back to the Local System instead of being saved on the Remote System. The default value of this variable is `False`. Setting this value to `True` indicates that the files listed on this variable should be transferred from the remote system to the save files directory on the local system.

**Note:** these variables will be given a suffix of ‘`TET_REMnnn`’ when written to the relevant configuration file, where `nnn` is the Remote System number assigned by the User.

## 9.2.4 Remote: Clean Mode

This tab allows you to specify variables that will be used by TETware in Clean mode.

**TET\_CLEAN\_TOOL** This variable enables you to specify a tool that TETware should use to clean test cases.

**TET\_CLEAN\_FILE** This variable allows you to specify additional command line arguments that can be passed to the clean tool specified by `TET_CLEAN_TOOL`. Please note: no validation is performed by the TETware GUI for the relationship between `TET_CLEAN_TOOL` and `TET_CLEAN_FILE`, it is the Users responsibility to ensure that they interact correctly.

**TET\_OUTPUT\_CAPTURE** These radio buttons indicate to TETware whether the standard output and error streams from the Clean tool should be captured and placed in the Journal file. The default value of this variable is `True`. Setting this value to `False` indicates that the standard output and the standard error output from Test Cases should not be captured and placed in the journal file.

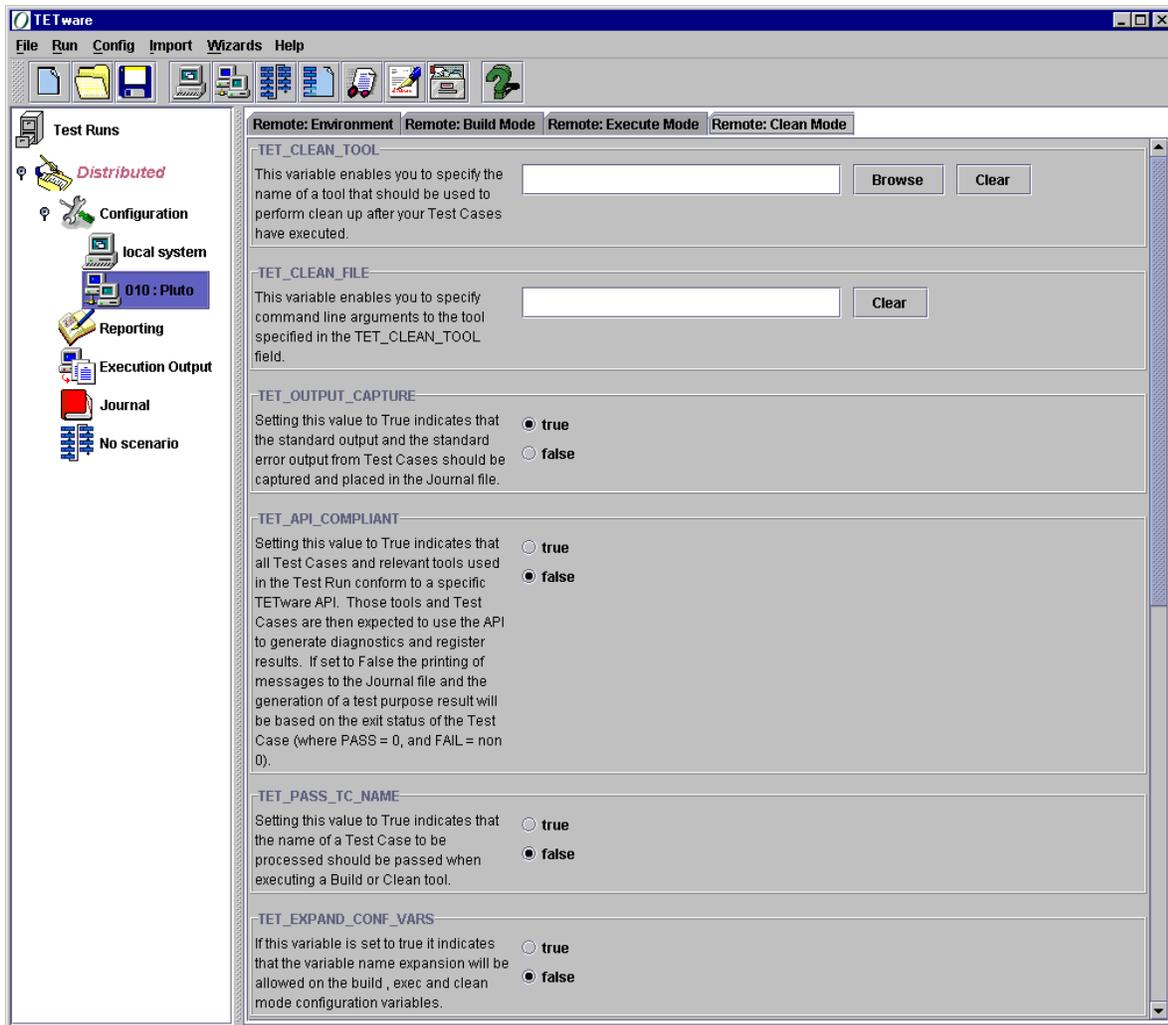


Figure 21: Remote: Clean Mode

**TET\_API\_COMPLIANT** These radio buttons allow you to specify whether the Clean tool and Test Cases to be built conform to the TETware API, i.e. they use the TETware API to output information to the Journal file. The default value of this variable is `False`. Setting the value to `TRUE` indicates that all Test Cases and relevant tools used in the Test Run conform to a specific TETware API. Those tools and Test Cases are then expected to use the API to generate diagnostics and register results. If set to `FALSE` the printing of messages to the journal file and the generation of the test purpose result will be based on the exit status of the Test Case (where `PASS = 0`, and `FAIL = non 0`).

**TET\_PASS\_TC\_NAME** These radio buttons allow you to specify to TETware whether the name of the current Test Case should be passed to the Clean tool specified. The default value of this variable is `FALSE`. Setting this value to `TRUE` indicates that the name of a Test Case to be processed should be passed when executing a Build or Clean tool.

**Note:** these variables will be given a suffix of `'TET_REMnnn'` when written to the relevant configuration file, where `nnn` is the Remote System number assigned by the User.

**TET\_EXPAND\_CONF\_VARS** The value of this variable determines whether or not `tcc` performs configuration variable expansion on variables defined in the per-mode configurations. If `true`, `tcc` replaces the string `${variable-name}` in a configuration variable assignment with the value of the variable `variable-name`. If `false`, `tcc` does not treat the string `${variable-name}` in a configuration variable assignment specially. The default value is `false`. For further information on this feature is presented in Section 5.8 of the TETware Programmers Guide.

## 9.3 Importing Configuration Files

The TETware GUI allows the import of build, execute and clean mode configuration files to the relevant tab and fields for the configuration of the Remote System highlighted in the tree. This feature makes it easier to configure a Remote System from the local machine.

### 9.3.1 Importing a Build Mode File

There are two ways to import a build mode configuration file:

1. When the Remote System is highlighted, select from the Menu: `Import / Import Build mode file`, a File Chooser will be shown. Select the file you want to import to the Remote System, then click on the `Open` button, the configuration value will appear in the relevant field under the `Remote: Build Mode` tab.
2. Right-click the Remote System tree node, a pop-up menu will be shown. Select `Import Build mode file`, then follow the same procedure as above.

## 9.3.2 Importing an Execute Mode File

There are two ways to import a execute mode configuration file:

1. When the `Remote System` is highlighted, select from the Menu: `Import / Import Execute mode file`, a File Chooser will be shown. Select the file you want to import to the Remote System, then click on the `Open` button, the configuration value will appear in the relevant field under the `Remote: Execute Mode` tab.
2. Right-click the `Remote System` tree node, a pop-up menu will be shown. Select `Import Execute mode file`, then follow the same procedure as above.

## 9.3.3 Importing a Clean Mode File

There are two ways to import a clean mode configuration file:

1. When the `Remote System` is highlighted, select from the Menu: `Import / Import Clean mode file`, a File Chooser will be shown. Select the file you want to import to the Remote System, then click on the `Open` button, the configuration value will appear in the relevant field under the `Remote: Clean Mode` tab.
2. Right-click the `Remote System` tree node, a pop-up menu will be shown. Select `Import Clean mode file`, then follow the same procedure as above.

## 2.1.1 Importing a Distributed Config File

There are two ways to import a Distributed Config File:

1. Highlight the **Local** System, select from the Menu: `Import / Import Distributed Config file`, a File Chooser will be shown. Select the file you want to import, then click on the `Open` button, the configuration value will appear in the relevant field under the.

2. Right-click on the **Configuration** tree node, a pop-up menu will be shown. Select `Import Distributed Config file`, then follow the same procedure as above.

See also Section 7.3.2.

## 9.4 Copying a Remote System

The advantage that this feature offers is that it helps to save time as it copies all of the configuration settings of one Remote System to a new Remote System.

To Copy a Remote System, you will need to have created at least one Remote System first. You will then have to carry out the following steps

1. Right click on a previously created and configured Remote System.
2. Select the option `Copy Remote System`. A dialog box will come up giving you the option of entering a name for the new Remote System.
3. Enter the name of the new Remote System into the window, and then click on OK. If you click on OK, without entering the information required, you will be prompted by an error message `'No Remote System name given or specified'`.
4. If details are entered then you will be prompted to enter a Remote System number between 1 and 999. Enter the Remote System Number and click on OK to complete the Remote System copying. If number 0 is entered here it will give an error message: `'Error: System number 000 is reserved for the Local System'` or if a number greater than 999 was entered, it returns an error message `'Remote System number is invalid'`.

## 9.5 Removing a Remote System

There are two ways to remove a Remote System:

1. On the Remote System tree node, Right-click on the node, a pop-up menu will appear, select `'Remove Remote System'`.

2. Press the Delete key when the Remote System is highlighted.

You will then be prompted with a dialog box with the confirmation message: Confirm removal of Remote System [*Number*]: [*Name*]?

If the **Yes** button is clicked then all components of the Remote System chosen will be removed from the tree. If **No** is chosen then the Remote System will not be removed.

## 9.6 Rename Remote System

You can rename and/or renumber a Remote System by following the procedure below:

1. Right-click on a Remote System in the Test Run tree, a pop-up menu will be shown, one of the items is: Rename Remote System.
2. Select this menu item, a dialog box for entering new name will be displayed. Enter the name and click on the **OK** button. A new dialog box for entering the new Remote System number will be shown.
3. In the dialog box for the new number, enter the new system ID for the Remote System, and click on the **OK** button. A new dialog box to confirm the change will be shown. If the new number is invalid, for example, out of range of the Remote System numbers, an error message will be displayed to inform the user, and promote user to try again.
4. Once the confirmation is done, the new name and new number will replace the old name and old number of the Remote System, and the change will be reflected by the tree node label.

At any stage of the dialog box, the **Cancel** option will abort the operation.

## 9.7 Systems Files

These files are only used by the Distributed TETware GUI. They are not used by the Lite or Embedded versions.

### 9.7.1 The systems File

The systems file should be present in the TET\_ROOT directory on each system and is used by Distributed TETware to map the TETware system IDs to some information that can be used by the network interface to identify a machine.

An example systems file is supplied with the demonstration in

`tet-root/src/tet3/demo/systems`, which contains lines similar to the following:

```
# Example system file for demonstration
000 Local
001 Remote
```

The systems file is automatically generated in the TET\_ROOT directory of the Local System when a Test Run is starting to execute. For Remote Systems, the file will be automatically copied to the mapped TET\_ROOT directory of the Remote System, if the `Mapping` variable has been defined under the Remote: Environment tab. Otherwise, the TETware GUI will tell the user to manually copy the file to the right directory before the test run execution continues.

### 9.7.2 Makefiles

You must check the test case `makefiles` on each system to ensure that they will build the test cases correctly.

You may need to add the names of libraries to the `SYSLIBS` variable in order to resolve external function names used in the TETware GUI and API. For example, if you have built TETware to use the socket network interface, you may need to append `-lsocket` and `-lnsl` to this variable on some systems.

If you are running either part of the distributed demonstration on a Windows NT System, you will need to customize the file name suffix variables in the `makefiles` as well as the `SYSLIBS` variable. Details of the changes that you must make are presented in comments contained in the `makefiles`.



## 10 Reporting – Configure Report Writer & Database

Before any report can be generated from the Test Run, the TETware Report Writer and/or Database need to be configured.

### 10.1 Configuring the TETware Report Writer

Depending on what information you require in your report, you can decide to check or uncheck the boxes next to the relevant variables. Some boxes are required while others are optional.

#### **Journal:**

This specifies the path of the import file, i.e. the journal from which the report is to be generated (This file can also be browsed and selected). After a Test Run has been executed this field is automatically completed. If the Test Run is saved the Journal name that was current when the Test Run is saved is also saved.

#### **Output File:**

This specifies the name of the file to which you want the generated output to be written. (If this file is already in existence, then you can click on the `Browse` button to select the file). You must specify the suffix as well as the file name. Note that you should specify the path as well. If you do not the report will be included in your `TET_ROOT` directory.

#### **Format:**

This specifies which document format you will want for the generated report. Either:

<code>Html</code>	Results can be viewed in HTML Format or
<code>Text</code>	Results can be viewed in test (txt) Format

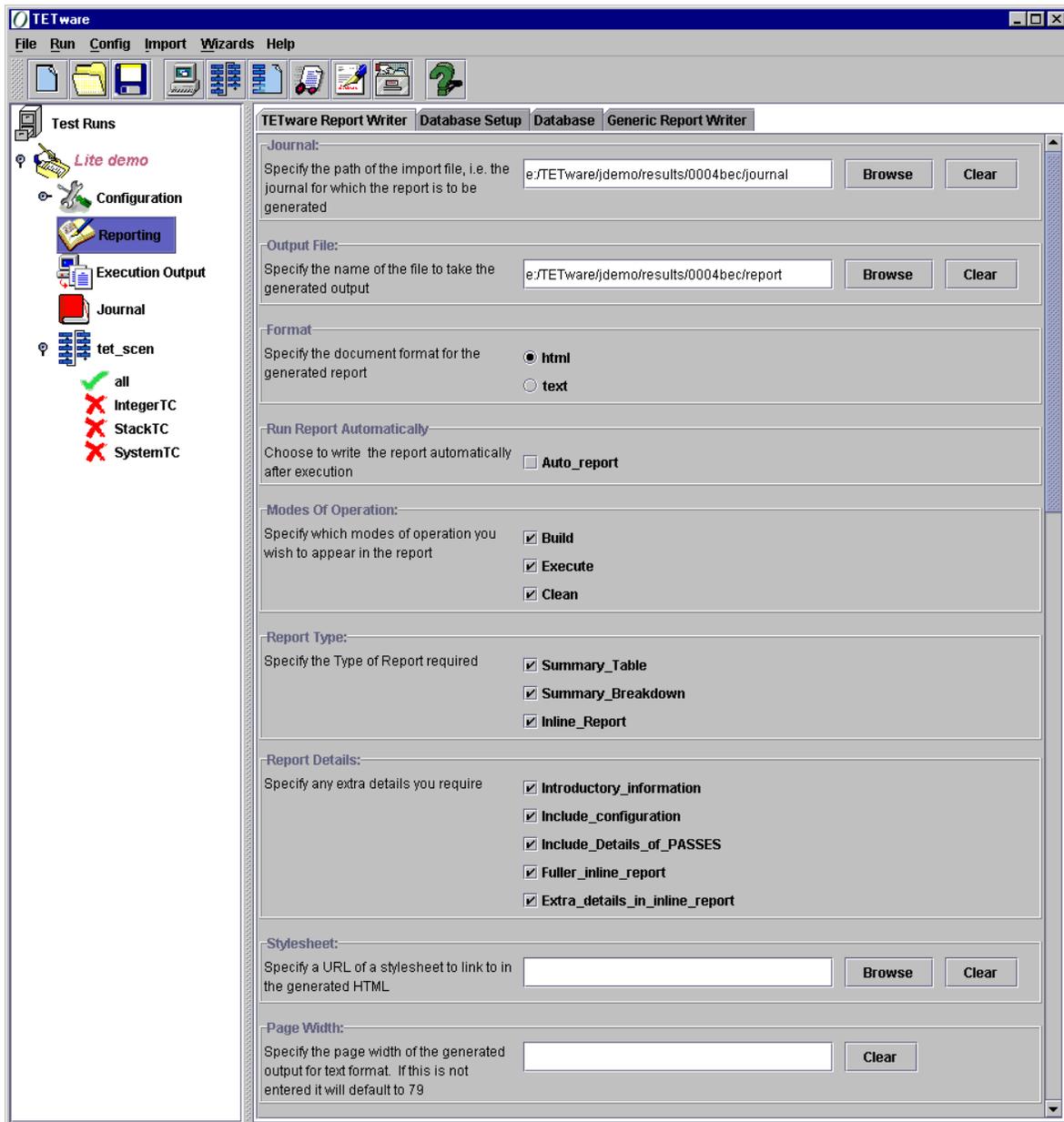


Figure 22: TETware Report Writer

## Run Report Automatically

If you want the journal file to be written as a report **automatically** after the test run execution is complete, then you should check the `Auto-report` checkbox. The Report will be written to the same directory as the journal file unless you specify a different path in the Output File field.

## Mode of Operation:

You will be able to specify which mode of operation should appear in the report. Any combination of the following can be chosen: Build, Execute or Clean. For more information on Modes of Operation please see Section 2.6.

## Report Type:

This allows the user to specify the type(s) of report required. User can pick a combination of the following: -

<b>Summary_table</b>	Provides a summary of the results by mode of operation.
<b>Summary_breakdown</b>	Provides a summary of the results by mode of operation and test case.
<b>Inline_report.</b>	Provides a summary of the results by test case and mode of operation.

## Report Details:

This allows you to specify any extra detail is required. Such details include:

- |                                       |  |
|---------------------------------------|--|
| <b>Include_Details of PASSES</b>      | If this option is selected details of tests that PASS will be included. Not selecting this item can help locate problems.                                    |
| <b>Introductory_information</b>       | This adds: the TETware version; System information; Date of Test Run and Start time to the top of your Report.   |
| <b>Fuller_inline_Report</b>           | This is an Inline Report broken down by Test Cases and Test Purpose (TP). You must have selected Report Type: Inline_Report for this option to be available. |
| <b>Extra_Details_in_inline_report</b> | This adds Scenario information to the Inline Report. You must have selected Report Type: Inline_Report for this option to be available.                      |
| <b>Include_configuration</b>          | This adds configuration information, by variable and value, to the top of the report (after introductory information).                                       |

## Stylesheet:

This option or field applies only if the html format has been selected. If specified the URL of the stylesheet to link to in the generated HTML.

## Page Width:

This option applies only to text format reports. It specifies the page width of the generated output. If this is not entered, it will use the default - which is 79 - will be used.

### 10.1.1 Creating a Report

Select the Run the TETware Report Writer button on the toolbar.



You can then go and view your report at the appropriate location using a tool that is appropriate for the format selected, e.g. your browser if the Report is saved in HTML format.

## 10.2 Configuring the Database Extraction tool

To be able to extract journal file parameters to the Database, some variables need to be set on the Database Setup and Database tabs.

### 10.2.1 Database Setup tab

For details of the Administrative aspects of Database Setup please see Section 3.4.

The TETware GUI Database Setup on Windows, UNIX and Linux Platform are all the same. It can be varied each time a Test Run is configured.

#### 10.2.1.1 Database Connection:

This refers to the Database connection being used. Consideration is given to the way in which the TETware GUI communicates with the users Database.

In Figure 23 `jdbc:odbc:TETbase` was used to configure a connection to a Microsoft Access database. Some other examples that could be used depending on the Database the individual is connecting to includes:

```

MYSQL      jdbc:mysql://localhost/TETbase
ORACLE     jdbc:oracle:thin@localhost:1521:TETbase
INFORMIX   jdbc:Informix:apdirect:system=tetsystem=TETbase etc.

```

Where *TETbase* is the name of the database file.

### 10.2.1.2 Database ODBC / JDBC Driver:

This field refers to the Database Links that are being used. In Figure 23, we have used the Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) Drivers `Sun.jdbc.odbc.jdbcOdbcDriver` to configure a connection to a Microsoft Access database. Some other examples that may be used include:

<b>MYSQL</b>	<code>org.gjt.mm.mysql.Driver</code>
<b>ORACLE</b>	<code>oracle.jdbc.driver.OracleDriver</code>
<b>INFORMIX</b>	<code>Informix.api.jdbc.JDBCdriver</code> <i>etc.</i>

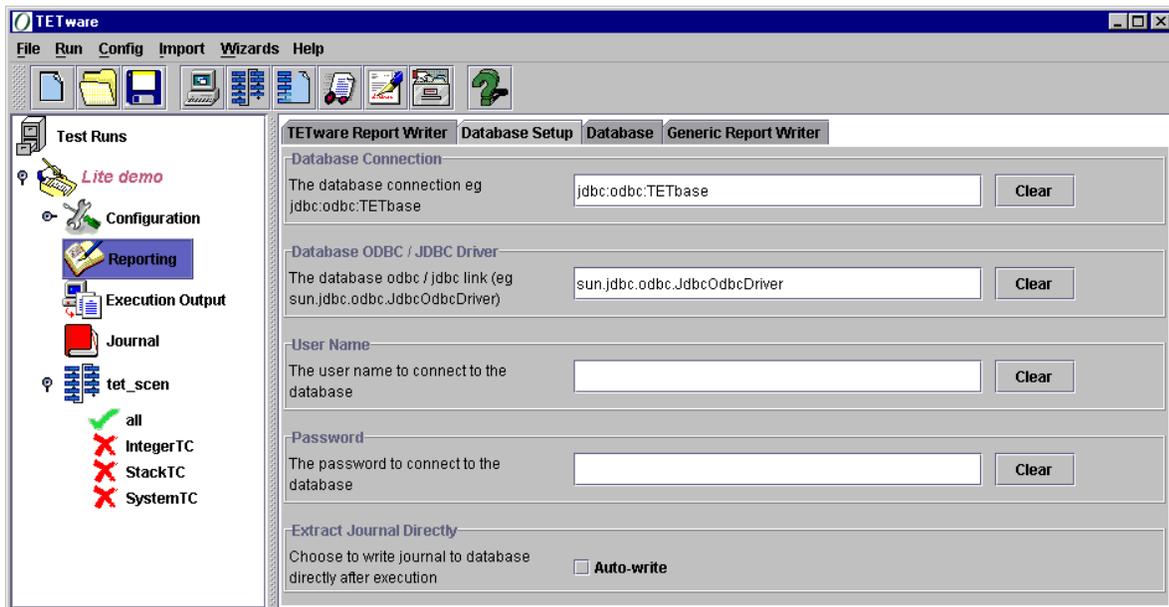


Figure 23: Database Setup tab

## 10.2.2 Remote Database:

If you are connected to a Database on a different system, you may be required by that Database system's security to supply a 'User's Name' and a 'Password'. This can be entered here so that it is not required every time a data connection is made.

### 10.2.2.1 User Name:

If your system requires a users name to access your Database file, then this users name has to be entered into this field.

### 10.2.2.2 Password:

If your system requires a password to access your Database file, then this password has to be entered into this field.

## 10.2.3 Extract Journal Directly

If you want the journal file to be extracted to the database **automatically** after the test run execution is complete, then you should check the `Auto-write` checkbox.

## 10.2.4 Database

Once the Database Setup is completed the user must select the Database tab, within the Reporting tree node, and then select the journal file parameters that they wish to pass to their database. These parameters are grouped according to the destination table in the database (please see Appendix B). Note that not all parameters or database fields are listed here. The key fields in each database table are mandatory and the relevant journal file parameters will be written to these fields each time the TETware GUI Database Extract tool is run.

### 10.2.4.1 Mode of Operation:

You will need to select the modes of operation for which you wish to pass Journal File parameters to the database. The options available are Build, Execute and Clean.

One or more of these options must be selected if you want to pass results to the Test Purpose table. If no Mode of Operation is selected the relevant parameters will only be passed to the Test Run and Configuration Variables tables.

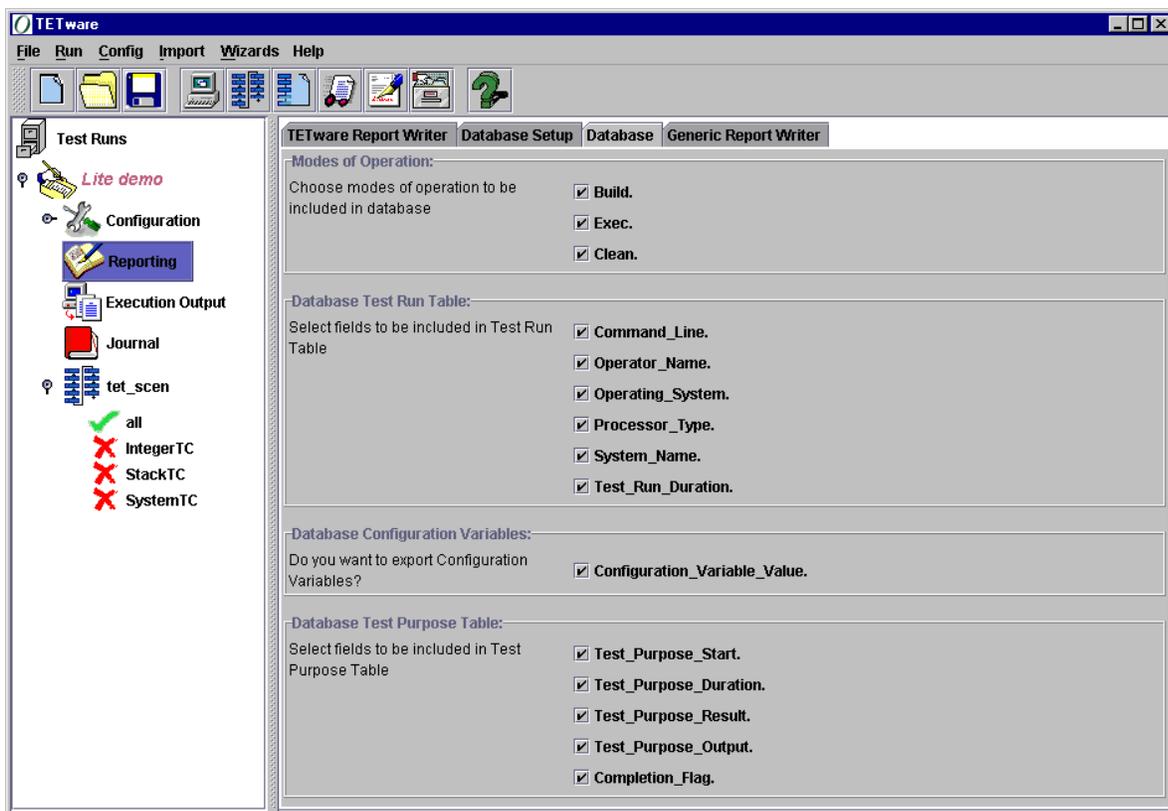


Figure 24: Database Tab

### 10.2.4.2 Database Test Run Table:

This option allows users to select fields to be included in Test Run Table. Available options are:

<b>Command Line:</b>	The command line used to execute the test.
<b>Operators Name:</b>	The name of the person who is logged on for this Test Run
<b>Operating System:</b>	The operating system resident on the system.
<b>Processor Type:</b>	The type of processor that the system is using.
<b>System Name:</b>	The name of the system that is being used.
<b>Test_Run_Duration:</b>	The time used to run the test

#### 10.2.4.3 Database Configuration Variables:

Selecting this field causes the configuration variables to be exported to the Configuration Variables table.

#### 10.2.4.4 Database Test Purpose Table:

This option allows users to select fields to be included in Test Purpose Table. Available options are:

<b>Test_Purpose_Start:</b>	The start time of the test purpose.
<b>Test_Purpose_Duration:</b>	The time taken to complete the test purpose.
<b>Test_Purpose_Result:</b>	The result code for this test purpose.
<b>Test_Purpose_Output:</b>	Output from this invocation of the test purpose.
<b>Completion_Flag:</b>	Flag indicating tests that failed to complete.

## 10.3 Configuring your own Report Writer

It is possible to generate reports for a Test Run using your own Report Writer.

The following criteria must be met for a report to be successfully generated for a Test Run:

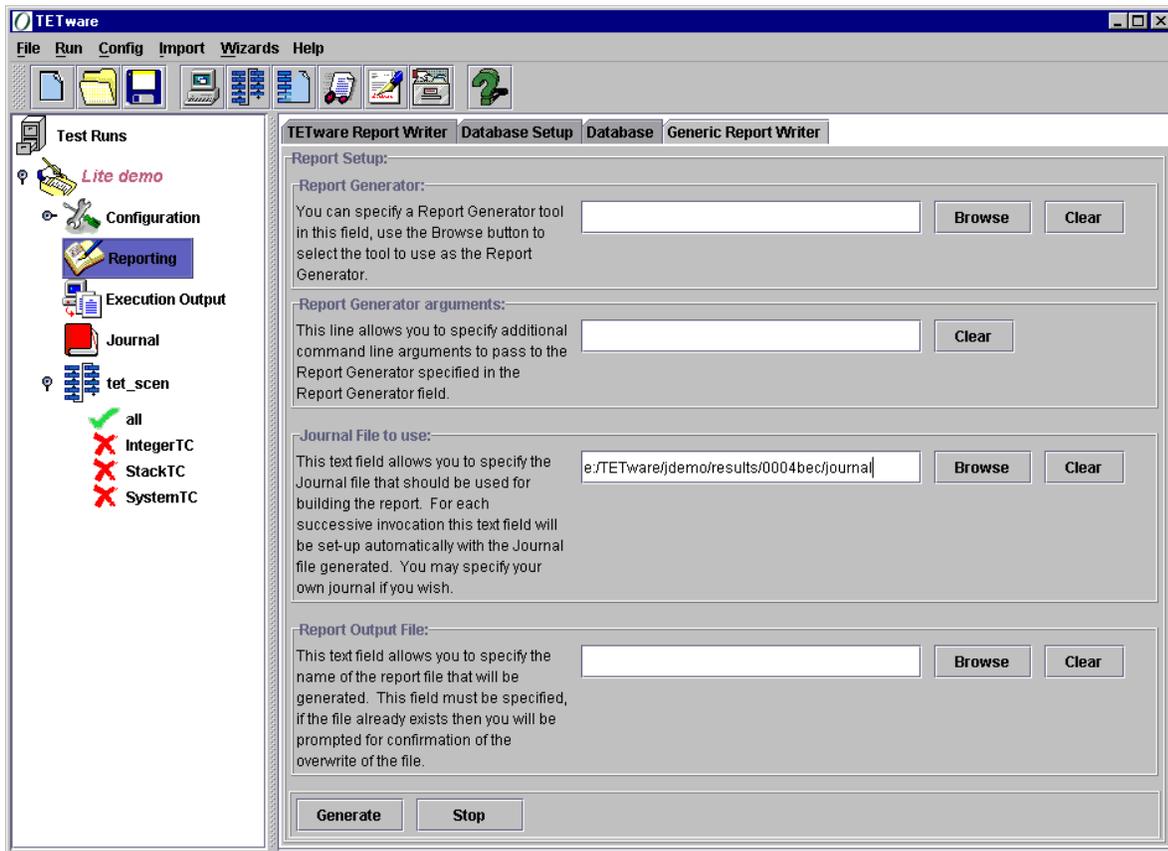


Figure 25: Generic Report Writer

**Report Generator:**

A valid Report Writer must be specified in this text field, the file system can be browsed to find the required Report Writer.

**Report Generator Argument:**

The Report Writer must take the name of the Journal file as an argument.

**Journal File to use:**

The Report Writer must send it's output to the standard output stream. This is because the output is captured by the TETware GUI and written to the output file.

**Report Output File:**

An output file must be specified in the Report Output File text field. The file system can be browsed to find an existing file, if an existing file is used then you will be prompted to overwrite the file. If no output file is specified then Report Generator Failed dialog box is displayed: with the message Error: The output file must be specified.



## 11 Executing a Test Run

To start a Test Run, you need to either:

- or right-click on the icon representing Test Run 
  - or use the 'Execute Test Run' menu item in the 'Run' menu
  - or use the keyboard shortcut Alt-E

There are some things that you need to be bear in mind first.

### 11.1 Pre-execution Criteria

To successfully execute a Test Run, a number of criteria that **MUST** be met, they are:

**TET\_ROOT must be specified for the Local System:** Without TET\_ROOT specified TETware is unable to find important library files and certain configuration files. See Section 7.2.1.

**TET\_SUITE\_ROOT should be specified:** The rules applicable to TET\_SUITE\_ROOT are as follows:

- If the path to the Scenario file lies on the same path as TET\_ROOT then TET\_SUITE\_ROOT does **NOT** need to be specified.
- If the path to the Scenario file does **NOT** lie on the same path as TET\_ROOT then TET\_SUITE\_ROOT **MUST** be specified. i.e. if the Scenario file path is /home/testsuite/tests/scen1.scn and TET\_ROOT is /home/tetware, then TET\_SUITE\_ROOT must be specified as /home/testsuite.
- If TET\_SUITE\_ROOT is specified then the Scenario file must lie on the same path. i.e. if TET\_SUITE\_ROOT is /home/testsuite then the path to the Scenario file must start with /home/testsuite.

For further details please see Section 7.2.1.

**At least one mode of operation MUST be specified**

TETware needs to know what you want to do before it starts working! (See Section 7.1.1).

**All Environment variables for the Local System MUST be directories**

The Environment variables specified for the Local System MUST be directories since they are locations where TETware will either look for files or place files. (See Section 7.2.1).

**Alternate configuration files must exist**

If the User specifies any alternative configuration files then they must already exist. (See Section 7.1.6).

If distributed testing is to be performed (i.e. Remote Systems were added to the Test Run), then the following criteria MUST be met for each Remote System:

TET\_ROOT must be specified for each Remote System (see Section 9.2.1).

TET\_TSROOT must be specified for each Remote System (see Section 9.2.1).

**One or more scenarios MUST be selected**

Testing cannot commence unless the TETware knows what it should be testing, so at least one Scenario must be selected from the Scenario file (see Section 8.3).

## 11.2 Starting Execution of a Test Run

After a Test Run has been selected, a scenario file picked and Test Cases are assigned, then a Test Run can be executed.

This is simply done by either of the three ways listed below:

- Click on the icon representing the Test Run on the Toolbar 
- Use the `Execute Test Run` menu item in the Run menu
- Use the Keyboard Shortcut Alt-E

Whichever way you choose to execute the Test Run, you will be prompted with a dialog which asks you to confirm start of execution for Test Run: [*Name of Test Run*]. At this point you have the option to vary the Mode of Operation from those determined on the Main Configuration tab (see Section 7.1.1).

If the **Yes** button is clicked then the TETware GUI will perform all the necessary checks to ensure that the Test Run can be executed, and if these are passed the Test Run Execution will begin. If any of the criteria outlined above are **NOT** met by the configuration for the Test Run then an appropriate dialog box will be displayed indicating the error (the dialog boxes are not displayed here due to the large number possible).

If all the criteria outlined above **ARE** met by the configuration for the Test Run then the TETware GUI will then generate all the configuration files that are needed by TETware to perform a successful execution (please see Section 11.3).

If **No** is chosen then the Test Run will be aborted.

When a Test Run begins to execute, the Display Window will automatically switch to the Execution Output window and the tree icon will change to . Once execution is complete the tree icon will change back to .

## 11.3 Execution Configuration Files

If all the criteria outlined above **ARE** met by the configuration for the Test Run then the TETware GUI will then generate the following configuration files that are needed by TETware to perform a successful execution:

### **`_GUItetbuild.cfg`**

This file is the build mode file for the Test Run, it is only generated if the User has selected Build mode as a mode of operation. The location of this file is dependent on the values of `TET_ROOT` and `TET_SUITE_ROOT`, if `TET_ROOT` is set then the file will be placed under it, if `TET_SUITE_ROOT` then it will be placed under there.

### **`_GUItetexec.cfg`**

This file is the execute mode file for the Test Run, it is only generated if the User has selected Execute mode as a mode of operation. The location of this file is dependent on the values of `TET_ROOT` and `TET_SUITE_ROOT`, if `TET_ROOT` is set then the file will be placed under it, if `TET_SUITE_ROOT` then it will be placed under there.

### **`_GUItetclean.cfg`**

This file is the clean mode file for the Test Run, it is only generated if the User has selected Clean mode as a mode of operation. The location of this file is dependent on the values of `TET_ROOT` and

TET\_SUITE\_ROOT, if TET\_ROOT is set then the file will be placed under it, if TET\_SUITE\_ROOT then it will be placed under there.

### **`_scenTETfile`**

This is a scenario file that is generated by the TETware GUI based upon the scenarios chosen. The file contains a TETware GUI generated scenario: `_TETallscen`, this scenario contains the list of scenarios chosen by the User. In this way the chosen scenarios can be executed by the TETware GUI. The location of this file is dependent on the values of TET\_ROOT and TET\_SUITE\_ROOT, if TET\_ROOT is set then the file will be placed under it, if TET\_SUITE\_ROOT then it will be placed under there.

**Please note:** this file is temporary and will be removed after the Test Run has finished execution, it is mentioned here for completeness.

### **`tet_code`**

This file is only generated if the User specifies Custom Result Codes, (see Section 7.1.5). The location of this file is dependent on the values of TET\_ROOT and TET\_SUITE\_ROOT, if TET\_ROOT is set then the file will be placed under it, if TET\_SUITE\_ROOT then it will be placed under there.

### **Systems**

This file is only generated if Distributed TETware is used and remote systems have been added to the Test Run. The location of this file is always in the TET\_ROOT directory.

### **`tetdist.cfg`**

This file is only generated if Distributed TETware is used and remote systems have been added to the Test Run. The location of this file is dependent on the values of TET\_ROOT and TET\_SUITE\_ROOT, if TET\_ROOT is set then the file will be placed under it, if TET\_SUITE\_ROOT then it will be placed under there.

**Please note:** It should be noted that for the same Test Run conditions the files are overwritten when the Test Run is re-executed.

Once all the relevant files have been generated the command line used to execute TETware is built. Once the command line has been built TETware is started with the environment set to the values indicated in the Test Run configuration.

**Please note:** a 2 second delay has been added to allow time for the Operating System to deliver the captured output to the TETware GUI, it was found during testing that there are

times when the TETware GUI can finish execution but the Operating System did not deliver the captured output before the TETware GUI had detected that it had finished. The consequence of this was that the TETware GUI indicated that it had finished but no output was displayed. To circumvent this a delay was added to enable the output to be captured correctly.

## 11.4 Quick Re-Run

When a Test Run has completed if there are some Test Cases that have failed to execute, the failed set of tests may be substantially smaller than the total number of Test Cases in the Test Run. Therefore, it may be desirable to re-execute only part of the Test Run, for example, the failed test cases.

The ability to re-execute quickly only those tests that have a result code of `FAILED` and `UNTESTED` is provided. To access this function the User can:

- Use the `Quick Re-run` menu item in the `Run` menu.
- Right-click on the `Execution Output` tree node, a pop-up menu appears, then click the `Quick Re-run` menu item.
- Use the Keyboard Shortcut `Alt-Q`.

**Note:** These menu items will only be available when the Test Run is NOT executing.

Whichever way you choose for the `Quick Re-Run`, you will be prompted with a dialog box with the message `'Confirm Re-run of Test Run [Name]`.

If the `Yes` button is clicked then a `Quick Re-Run Execution` will begin.

If `No` is chosen then the Test Run will be aborted.

Please note that the process of re-executing of a Test Run is the same as for normal execution, this means that all the rules and criteria applicable to normal execution also apply for the `Quick Re-run`. If you have changed any configuration values for the Test Run since it was executed normally then this will have an effect on the values written to the Journal file.

## 11.5 Advanced Re-Run

There may be times when the User wishes to re-execute a Test Run but wants to have more control over what is re-executed, e.g. they may wish to re-execute all tests that have a result code of NORESULT. To do this the 'Advanced Re-run' function is provided, to use this function the User can:

- Use the `Advanced Re-run` menu item in the `Run` menu.
- Right-click on the `Execution Output` tree node and click the `Advanced Re-run` menu item from the pop-up menu.
- Use the Keyboard Shortcut `Alt-V`

**Note:** these menu items will only be available when the Test Run is NOT executing.

Whichever method is chosen a dialog box will prompt you to choose between two options namely

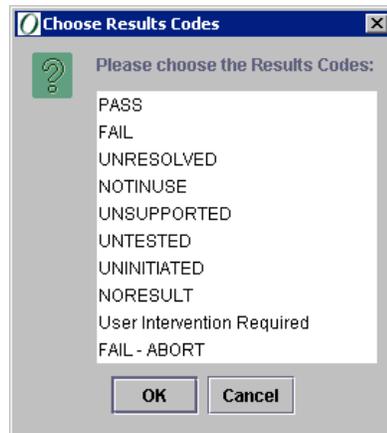
**Results Codes:** Here the User can specify the results codes that they wish to use for the re-run. This is the default option.

**Modes of Operation:** Here the User can specify which Modes of Operation they wish to use for the re-run.

### 11.5.1 Choosing Results Codes

If you choose `Results Codes` and click on `OK` then a dialog box with standard `Results Codes` together with user defined `Results Codes` will be displayed, as shown in Figure 26.

You will then be able to choose one or more `Results Codes` that you wish to use for the Re-run. To select a contiguous list of `Results Codes` you should hold '`Shift`' and click the desired list of codes. To select a dis-contiguous list of `Results Codes` you should hold '`Ctrl`' and select the list of codes.

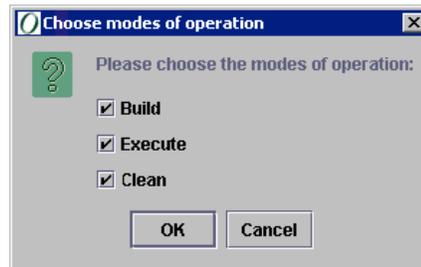


**Figure 26: Choose Results Codes**

Once you are satisfied with the Results Codes chosen, you should click OK, the Test Run will be then be Re-run for the Results Codes specified with the configuration present in the Test Run. Select Cancel to abort.

### 11.5.2 Choosing Modes of Operation

If you choose Modes of Operation and click OK then the following dialog box will be displayed



**Figure 27: Choose modes of operation.**

This dialog box allows you to specify the Modes of Operation that you wish to use in performing the re-run. You are able to specify one or more Modes of Operation. Note, at least one Mode of Operation MUST be specified to continue.

If one or more Modes of Operation are specified and OK is chosen then the Test Run will be Re-run for the modes specified with the configuration present in the Test Run.

If no Mode of Operation is specified and OK is chosen then an error dialog box with the message 'Error: No modes of operation chosen, aborting' will be displayed.

## 11.6 Stop Test Run.

**Please note:** Stopping a Test Run whilst it is executing is NOT recommended and should be avoided where possible due to the potentially undesirable effects it can have on the system.

To stop a currently executing Test Run the User can:

- Use the Stop Test Run menu item in the Run menu.
- Right-click the Execution Output tree node and click the Stop Test Run menu item.

**Note:** these menu items will only be available if the Test Run is currently executing.

Whichever method is chosen a dialog box will be displayed with the message 'Confirm halt of execution of New Test Run'. It further asks you if you are sure you wish to continue to stop the Test Run as stopping it may have an un-desirable impact on the system.

If you choose Yes then the Test Run will be stopped and the Execution Output tree node icon

changed to . However if No is clicked, the Test Run will continue.

**Note:** It may take some time for the process to stop executing, this is due to TETware trying to perform some clean up before stopping. Please be patient when stopping Test Runs.

## 11.7 Execution Output

When a Test Run is been running, the icon of Execution Output tree node will change from  to ; and it will change back to  once the execution is completed.

Whilst executing a Test Run the TETware GUI will capture the standard output and standard error streams and display them in the "Execution Output" and "Error Output" text areas on the right-hand side of the TETware GUI respectively.

Figure 28 shows a Test Run that has finished executing:

### 11.7.1 Execution Output

This window will display the path and name of the journal file to which the Output was written. It will also display the time, modes of operation and the Test Cases used during the Test Run.

### 11.7.2 Error Output

This window will display any error messages for you after you have executed a Test Run. If you did not get the right Output from your Test Run, you should check in this box for any error messages and make the necessary corrections in the TETware GUI. If you need to contact us for support, you will also need to include the message you have in this field along with your email and any Journal File. Highlight the Error Output text and use `Ctrl+C` to copy it into an email.

### 11.7.3 Journal File

Once the output has been captured the TETware GUI will determine what the Journal file name is for this Test Run execution. The Journal file name once captured will be used to display the contents of the Journal file in the Journal text area. Please see Section 12 for more details.

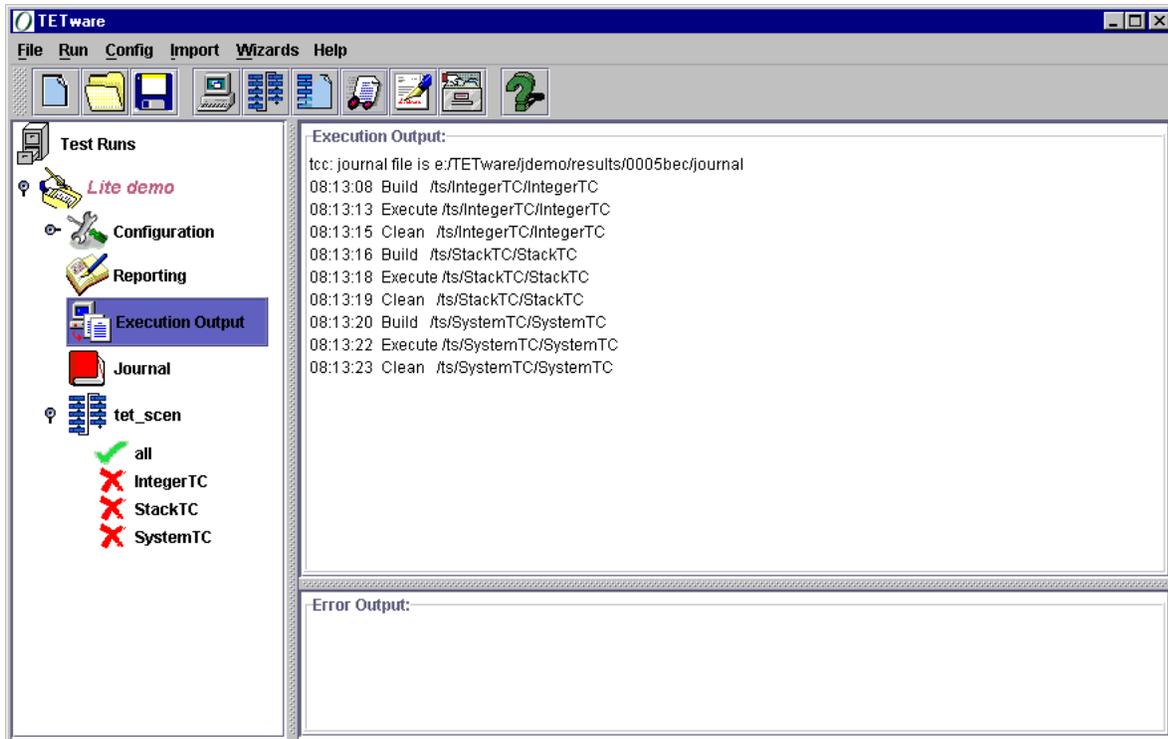


Figure 28: Execution Output

## 11.8 Test Case Locking

When TETware processes a test case, it prevents possible interference from other instances of tests by acquiring one or more exclusive or shared locks. These locks are acquired in the test case source directory and (optionally) in the test case execution directory. An exclusive lock is acquired by creating a file called `tet_lock` in the directory that is to be locked.

A shared lock is acquired by creating a file with a unique name below a directory called `tet_lock` in the directory that is to be locked, creating the `tet_lock` directory if necessary as well. A diagnostic is printed to the journal file if TETware is unable to acquire the locks that it needs. Although each instance of this tests is always careful to remove the locks that it created when they are no longer required, it is possible that old locks might remain after some system or process malfunction. In the event of this happening it is necessary to remove the old locks before proceeding (the locks are always located in the directory that contains your test cases). For further information see Section 3.10 of the TETware Programmers Guide.

## 12 Journal

The journal facility provides a mechanism for outputting data to the execution results file. The API ensures that each entry in this file is written automatically and that there is sequencing information applied to the entry.

### 12.1 Generating Journal Output

Journal files are automatically generated and stored in sequence under the directory `test_suite_root\results\NNNN[bec]`. (Where *NNNN* represent the sequence number allocated by the program and 'b e c' represents the mode of operation: build, execute and clean.

### 12.2 Viewing a Journal File

If a Test Run has been executed and a Journal file created by TETware then that Journal file can be viewed by clicking on the name `Journal` on the Tree Node. The Journal file will be parsed and placed into the `Journal Text` area. This behavior is automatic.



## 13 Reporting

### 13.1 Generating a Report with the TETware Report Writer

If you want to Generate a Report, Select the Run the TETware Report Writer button on the toolbar  or use the Keyboard Shortcut Alt-G.

You can then go and view your Report at the specified location (see Section 10.1) using a tool that is appropriate for the format selected.

### 13.2 Database Extract

If the Auto-write option is selected on the Database Setup tab (see Section 10.2.3) and all the functions have been set up properly (see Section 10.2) the Journal file parameters will be written to the database as soon as the Journal file is posted by the TETware GUI.

Alternatively you can cause the Journal file parameters to be written to the database by:

1. Clicking on the toolbar icon  that starts the extraction of the journal file parameters to the Database or using the Keyboard Shortcut Alt-D.
2. The Open window will come up to give you the option of selecting the journal file you wish to extract into the Database. (By default, it will come up with the last run journal file created). Then Click on Open . You may however click on the Cancel button to abort the operation.
3. If all settings and all parameters are right then the Database message box 'Database Extract Finished' will appear.
4. If an error message appears with the message 'Problem with action - There is no Test Run Active'. Please go back to the TETware GUI main screen and make a Test Run active by clicking on any of the options on the left side of your screen. Then start the process again.

5. If you select an invalid file or path. It will come up with an error message 'Extract problem – (The specified path is invalid). You will then need to go back and select a valid path and file.

### **13.3 Generating your own Report:**

It is possible to generate reports for a Test Run using your own Report Writer. See Section 10.3 for Setup instructions.

If the Test Run has been successfully executed then the name of the Journal file used will be pre-filled in the 'Journal file to use' text field, you can change this file if they wish and browse the file system for a replacement. Please note: the Journal file **MUST** exist, if not an error dialog, Journal file does not exist, will be displayed.

If all the criteria outlined above are met then press the `Generate` button to start the production of a report. When the Report writer has finished and the report generated then the `Report Generation finished` dialog box will be displayed.

You can stop the Report writer at any time by clicking `Stop`.

## 14 Saving a Test Run

There are four ways to save a Test Run:

- Use the  icon on the toolbar
- Use the 'Save' or 'Save As' menu items from the File menu
- Right-click on the Test Runs tree node and click the 'Save' or 'Save As' menu items.
- Use the Keyboard Shortcuts Alt-S (Save) or Alt-A (Save As).

### 14.1 Using the 'Save' option

You can either use the icon on the toolbars, or the 'Save' menu item from the File menu, or use the 'Save' menu by right-clicking on the Test Run tree node, or use the keyboard shortcut Alt-S, you will be prompted with a File selection dialog box.

The file selection dialog box will let you traverse the file system. Once the appropriate file has been chosen and the Save button clicked, the TETware GUI will attempt to save the Test Run to the chosen file<sup>4</sup>. If the chosen file already exists, then an Overwrite warning dialog box will be displayed.

If you choose Yes, the existing file will be completely overwritten and the Test Run saved to that file instead. If the file does not exist then the chosen file will be created in the file system and the Test Run written to that file.

If you choose No the Test Run will not be saved.

Each Test Run that is saved is given an extension of .trf, this is for convenience only, and the user is able to change the extension of the Test Run file. A file filter of \*.trf is provided in the file selection box to allow you to find Test Run files easily and quickly. If the extension is changed, it may then become difficult to find in the future.

---

<sup>4</sup> Please note: the Test Run is written to the chosen file in XML format. Any editing of the Test Run file may cause that Test Run to be unusable. The Open Group can accept no responsibility for Test Run files that have been edited by the User. If the 'Save file' variables has been set, i.e. by saving the Test Run at any time, then when the 'Save' menu item or toolbar icon is clicked then the Test Run will be saved quietly to the chosen save file.

## 14.2 Using the 'Save As' Option

Using the 'Save As' function allows you to save the Test Run as any name you wish at any time, regardless of whether the 'Save file' variable has been set for a Test Run. When the 'Save As' function is used, a file selection dialog is displayed. You will be able to browse the file system using the dialog box. Once the selection dialog box is displayed, the procedure is the same as for the 'Save' functions. (Please see Section 14.1 above).

## 14.3 Setting Defaults

It is possible to set the values that a Test Run contains as the defaults for that session of the TETware GUI. Once the defaults are set it is then possible to set the configuration values for another Test Run to those values.

It should be noted that setting the defaults for a Test Run is handled in different ways, depending upon whether you try to set the defaults via the "Config" menu or via a right-click on a node in the Test Run.

### 14.3.1 Config menu default setting

#### 14.3.1.1 Save as default settings

If the "Save as default settings" menu item is chosen from the "Config" menu then the TETware GUI will:

- Set ALL the configuration values in the main configuration (all tabs related to the "Configuration" tree node) as the defaults, this means that any values that are set as blank in the tabs will be saved as blank.
- Set ALL the configuration values in the local system as the defaults, again this means that any values that are set as blank in the tabs will be saved as blank.

It should be noted that if any remote systems have been added to the Test Run then they will NOT have their values saved as the defaults. This is due to the fact that to set the defaults for another Test Run that does not use any remote systems would not make any sense in terms of remote systems. If you wish to set the defaults for a remote system please see below.

### 14.3.1.2 Revert to default settings

If the "Revert to default settings" menu item is chosen from the "Config" menu then the TETware GUI will:

- Set the currently Active Test Runs main configuration to the saved defaults.
- Set the currently Active Test Runs local system configuration to the saved defaults.

It should be noted that if there are any remote systems present in the Test Run then the configuration values associated with that remote system will NOT be set.

## 14.3.2 Default setting via right-clicking on a tree node

### 14.3.2.1 Save as default settings

It is also possible to set the defaults for a Test Run via the right-click menus on the various tree nodes. It should be noted that when the defaults are set via the right-click menus then only the configuration associated with that tree node will be set as the defaults. The configuration that you can set the defaults for via the right-click menus are:

- The "Configuration" node, this will set the main configuration values for a Test Run as the defaults.
- The "local system" node, this will set the local systems values for a Test Run as the defaults.
- The "remote system" node, this will set the associated remote systems values as the defaults.

### 14.3.2.2 Revert to default settings

If the "Revert to default settings" menu item is chosen from a right-hand menu then the configuration values associated with that tree node will be set to the saved defaults.

**Please note:** If no defaults have been specified and the User chooses to set the defaults (regardless of origin) then the configuration values will be set to blank, there are NO built in default values coded into the TETware GUI defaults.



## 15. Wizards

The TETware GUI includes 3 wizards that are included to make the task of configuring Test Runs and Remote Systems easier. They do this by stepping the User through the various components of the configuration tabs in sequence. The wizards are useful because they move between the different tabs to ensure that all the appropriate options are considered. They are largely, self-explanatory.

The three Wizards are:

- Test Run Setup (Basic) Wizard
- Test Run Setup (Advanced) Wizard
- New Remote System Wizard

All three Wizards can be accessed via the Wizards menu, or using the Keyboard Shortcut Alt-W. Note that a test run must be open before the New Remote System Wizard can be run, and that this wizard is only available if you are using Distributed TETware.

Each Wizard is stored in a `Config` file that you will find in your `TET_ROOT` directory. These Config files are text files and may be opened using a text editor. The structure of the file is self-explanatory. Users may edit the Config file, but will need to store it under one of the existing names to be able to access it from the menu or toolbar (if you would like to be able to add your own wizards please email [tetware\\_manager@opengroup.org](mailto:tetware_manager@opengroup.org)). Please remember to back up the Config Files before editing them.

The three (3) Wizards provided are detailed in Tabular form below.

### 15.1 Test Run Setup (Basic) Wizard

This Wizard steps the User through the four minimum steps that **MUST** be carried out to configure a new Test Run. This Wizard can be accessed by:

- Selecting the Start New Test Run Wizard icon  on the toolbar
- Selecting the Wizards Menu, then Test Run Setup (Basic) Wizard
- Using the Keyboard Shortcut Alt-B

Label	Please See Section
Test Run Name	7.1.1
Test Run Description	7.1.1
TET_ROOT	7.2.1
Scenario File	8.2

**Table 10: Test Run Setup (Basic) Wizard**

## 15.2 Test Run Setup (Advanced) Wizard

This Wizard steps the User through ALL the steps that MUST and MIGHT be carried out to configure a New Test Run. There are 40 steps in this Wizard. The User can click on **Next** to pass through steps that are not relevant to their Test Run setup or Click on **FINISH** to end all setups.

This Wizard can be accessed by:

- Selecting the Wizards Menu, then Test Run Setup (Advanced) Wizard
- Using the Keyboard Shortcut Alt-Z

Label	Please See Section
Name	7.1.1
Test Run Description	7.1.1
TET_ROOT	7.2.1
Scenario File Selection	8.2
Modes of Operation	7.1.1
Build mode configuration file	7.1.6
Execute mode configuration file	7.1.6
Clean mode configuration file	7.1.6
Include Test Cases	7.1.4
Exclude Test Cases	7.1.4
Custom Result codes	7.1.5
Customize User Configuration Variables	7.1.7
Timeout	7.1.8.1
TET_XTI_TPI	7.1.2
TET_XTI_MODE	7.1.2
TET_LOCALHOST	7.1.2
Journal File Name	7.1.8.2
TET_SUITE_ROOT	7.2.1
TET_EXECUTE	7.2.1

TET_TMP_DIR	7.2.1
TET_RUN	7.2.1
TET_BUILD_TOOL	7.2.2
TET_BUILD_FILE	7.2.2
Build Mode TET_OUTPUT_CAPTURE	7.2.2
Build Mode TET_API_COMPLIANT	7.2.2
Build Mode TET_PASS_TC_NAME	7.2.2
Build Mode TET_EXPAND_CONFIG_VARS	7.2.2
TET_EXEC_TOOL	7.2.3
TET_EXEC_FILE	7.2.3
Execute Mode TET_OUTPUT_CAPTURE	7.2.3
Execute Mode TET_API_COMPLIANT	7.2.3
Execute Mode TET_EXEC_IN_PLACE	7.2.3
Execute Mode TET_SAVE_FILES	7.2.3
Execute Mode TET_EXPAND_CONFIG_VARS	7.2.3
TET_CLEAN_TOOL	7.2.4
TET_CLEAN_FILE	7.2.4
Clean Mode TET_OUTPUT_CAPTURE	7.2.4
Clean Mode TET_API_COMPLIANT	7.2.4
Clean Mode TET_PASS_TC_NAME	7.2.4
Clean Mode TET_EXPAND_CONFIG_VARS	7.2.4

**Table 11: Test Run Setup (Advanced) Wizard**

## 15.3 New Remote System Wizard

This Wizard steps the User through ALL the steps that MIGHT be carried out to start and configure a New Remote System. There is a total of 29 steps in this Wizard. The User can click on NEXT to pass through steps that are not relevant to their Remote System setup or Click on FINISH to end all setups.

This Wizard can be accessed by:

- Selecting the Start New Remote System Wizard icon  on the toolbar
- Selecting the Wizards Menu, then Start New Remote System Wizard
- Using the Keyboard Shortcut Alt-M

Label	Please See Section
Name	9.1
System Number	9.1
TET_ROOT	9.2.1
Mapping	9.2.1
TET_TSROOT	9.2.1
TET_SUITE_ROOT	9.2.1
TET_EXECUTE	9.2.1
TET_TMP_DIR	9.2.1
TET_RUN	9.2.1
TET_BUILD_TOOL	9.2.2
TET_BUILD_FILE	9.2.2
Build Mode TET_OUTPUT_CAPTURE	9.2.2
Build Mode TET_API_COMPLIANT	9.2.2
Build Mode TET_PASS_TC_NAME	9.2.2
Build Mode TET_EXPAND_CONFIG_VARS	9.2.2
TET_EXEC_TOOL	9.2.3
TET_EXEC_FILE	9.2.3
Execute Mode TET_OUTPUT_CAPTURE	9.2.3
Execute Mode TET_API_COMPLIANT	9.2.3
Execute Mode TET_EXEC_IN_PLACE	9.2.3
Execute Mode TET_SAVE_FILES	9.2.3
TET_TRANSFER_SAVE_FILES	9.2.3
Execute Mode TET_EXPAND_CONFIG_VARS	9.2.3
TET_CLEAN_TOOL	9.2.4
TET_CLEAN_FILE	9.2.4
Clean Mode TET_OUTPUT_CAPTURE	9.2.4
Clean Mode TET_API_COMPLIANT	9.2.4
Clean Mode TET_PASS_TC_NAME	9.2.4
Clean Mode TET_ EXPAND_CONFIG_VARS	9.2.4

**Table 12: New Remote System Wizard**

## 16 Help

### 16.1 TETware GUI Help Pages (In-Line Help)

All the TETware GUI tabs include help text on each item within the tab.

You can display help information on your screen at anytime by clicking on the Help icon,  or by choosing the Help index command from the Help menu, or by using the Keyboard Shortcut Alt-H.

Online Help for the TETware GUI is set up in such a way that the user will find it easy to move around and collect whatever information or assistance they may require.

### 16.2 TETworks Web Site (On-line Help)

TETware Support customers may also access the help files without opening the TETware GUI. Use your browser to access them at <http://tetworks.opengroup.org/> and log-on to the Support Pages.

### 16.3 How to use the Help Pages

The Help Pages are organized in such a way that you are able to access any part of the help pages at any time. The Tree structure employed enables you view at a high level any section of the help pages.

The left hand side of the help is a tree structure which represents the all the pages available in the help. The right hand side of the help is a Java HTML Editor which is used for display of the help pages. At present the HTML Editor only supports HTML3.2.

The icons used in the Help tree are detailed below:

Icon	Description
	This icon is a folder which will contain a number of pages of help.
	This icon is the closed version of the open folder for help pages.
	This icon indicates that the folder contains a number of help pages but also has a page of help associated with the folder itself. Clicking on the folder will display the associated page.
	This icon is the closed version of the open folder indicated above.
	This icon indicates that there is a page of help associated with this tree node. Clicking on the tree node will display that page of help.

Table 13: Help Icons

### 16.3.1 Using the Tree

Any node in the Help tree that has a page of help associated with it will have a  icon overlaid on it's folder icon or it's icon will be a , to access that page of information just click on the node. The help page will then be displayed in the right hand side of the help window.

To expand any folder in the tree just double click on the folder icon, it's contents (if any) will then be displayed.

### 16.3.2 Closing the Help window

If you close the Help window then it will disappear, if you then choose to browse the Web pages again via the "Help" menu in TETware then the Help window will be displayed again, but in the same state as when it was closed.

## 16.4 Creating Your Own Help Pages

**Note:** The ability to create your own help pages is NOT a supported function within the TETware GUI, it is provided for the Users benefit. The Open Group cannot accept responsibility for any adverse effects it may have on the TETware GUI installation. No problem reports will be accepted against the TETware GUI that relate to the production of Personal help pages.

There may be times when it is desirable to have supplementary information relating to the management of Test Runs recorded somewhere that the User may wish to refer to. It is possible to do so with the Help Tree. The Help window can display both plain text and HTML3.2.

It is recommended that if you wish to create your own HTML based Help Pages then you use the markup features used in the TETware GUI pre-written help pages.

The Help pages are created as follows:

- At start-up the TETware GUI reads in the `helpstructure.file` that should be present in the `TET_ROOT/gui/help` directory. The `TET_ROOT/gui` directory is the directory that the TETware GUI was installed to, it will look for the help structure file in the 'help' directory located off that directory.
- The `helpstructure.file` is parsed and the help tree built based upon the syntax in the file.

The syntax is outlined below:

Each valid line in the help structure file is composed of 3 parts. Any less than 3 parts and the line will be ignored. The parts are:

### **Identifier:**

The Identifier indicates to the help parser what the type of this line is, the types are described below, and the Identifier should be first thing that is present on the line:

**ROOT:** This indicates the start of the Help Tree structure. It is the starting point of the Tree, do not create more than one ROOT since the preceding ROOT and its sub tree nodes will be lost.

**H[value]** This is a heading value, with the [value] indicating the depth of the heading. Basically a heading is a folder for the Help Tree. Currently a depth of 1-9 is supported. A depth of 0 will probably cause a trap and the Help Tree not being built and hence displayed. Values greater than 9 will be interpreted as the first value in the number, i.e. 10 will be interpreted as 1, 31 as 3 etc.

**SUB** This is the indicator for a page of data that will have no children, if you create a heading below a SUB or list of SUBS then it will be added to the appropriate node above the SUB or SUBS indicated.

### **Name**

This is the name that you would like to be displayed in the Help Tree, it is basically the node name of the item. The name is also displayed in the Window title if you click on the node and there is a page associated with it.

### **Page**

This is the page that you wish to associated with the Tree node, the path to the page should be relative to the TET\_ROOT/gui/help directory, the path should be a full path specification, i.e. if the page is held in /TETware/help/MyHelp/my\_help\_page.html, then the page specification should be /MyHelp/my\_help\_page.html. The leading / is important. If you do not want a page to be associated with a given node then use the text 'null' instead of a page specification. For folders this will mean that no info indicator is overlaid on the folder icon. For info nodes it doesn't make sense to use 'null' but it is possible, in that case when the icon is clicked no page is displayed.

The Identifier and Name **MUST** be separated by a '=', if they are not then the line is considered invalid and ignored.

The Name and Page specification should be separated by a '|', again, if they are not then the line is considered invalid and ignored. No blank space should be placed between the '|' and the start of the Page specification as it will be considered part of the specification and switching to that page will not be possible.

The page given for ROOT (if present) is then displayed as the start page.

### **To create your own help pages follow the steps indicated below:**

1. **Create your Help Pages** You should create your help pages and place them in a sub directory from the TET\_ROOT/gui/help directory. You can create any depth of directory structure you need. The help pages can be plain text or in a HTML3.2 format.  
(Note: Currently Java 2 only supports HTML3.2, frames are NOT supported).

2. **Create a back-up copy of helpstructure.file** Create a back-up copy of helpstructure.file to ensure that you are able to undo any changes.
3. **Modify helpstructure.file** Modify helpstructure.file to include your help pages/folders etc.
4. **Reload the TETware GUI** Once you reload the TETware GUI your help pages should now be visible.

## Important Notes

The following are a list of important notes that should be read and followed whenever you create your own Help Pages:

### Beware of Images:

Image path specification can be a problem with the Java HTML editor. To specify the location of images to be displayed in the Help Page you should use the following format:

```
<IMG SRC='file:help/[yourpath]/yourimage.jpg'>
```

The syntax here is very important, the IMG SRC directive **MUST** start with 'file:help', no leading / should be placed before 'help'. GIF images (.gif) are also supported by the Editor.

- **Beware of Image Names:**

Image names can also be a problem, it has been observed that image names that are all capitals are NOT loaded into the page correctly (this problem has only been observed on Win32 systems). Where possible it is best to use all lower case names for images.

- **Rendering the <CENTER> Tag:**

It has been observed that help page written in HTML that contains the <CENTER> tag will not be rendered correctly. No reasonable explanation has been found for this, however a work-around using a <BR> tag is effective. Just use:

```
<CENTER>
desired HTML
<BR>
</CENTER>
```

This problem is seen when a node in the help tree is clicked but no associated page is displayed and the nodes color is not changed.

### 16.4.1 Examples

Here are a few examples for creating your own pages:

#### Creating a New Folder with no Associated Page with Sub-Pages

In general for completely new folders it is best to create a whole new top-level folder, so use:

```
HI=My folder|null
```

Here a new folder is created with no associated page. For sub pages it is best to create whole new directory under the `TET_ROOT/gui/help` directory, so create:

```
TET_ROOT/gui/help/My_folder/
```

To add sub pages use:

```
SUB=My sub page | /My_folder/my_sub_page.html
SUB=My other sub page | /My_folder/my_other_sub_page.txt
```

If your .html files contains images then create a /images directory from the My\_folder directory and place images in there. When referring to the images in the .html file use:

```
<IMG SRC='file:help/My_folder/images/my_image.jpg'>
```

## Creating Sub Folders

If you wish to create a sub folder from an existing one then you should create a new directory in the file system that branches from the folder you wish to branch from. So if folder `TET_ROOT/gui/help/Getting_started/` already exists and you wish to create a new folder under it then create a directory to hold the files:

```
TET_ROOT/gui/help/Getting_started/My_sub_folder/
```

Then place your help pages in the `My_sub_folder` directory. Modify `helpstructure` `.file` to add your sub folder, for sub folders you should add one to the number of the heading you wish to branch from, i.e. If the following already exists in `helpstructure` `.file`,

```
H1=Getting Started|null
```

Then, you add something like:

```
H2=My starting page|/Getting_started/My_sub_folder/my_starting_page.html
```

## 16.5 Feedback and Problem Reporting

If you have any comments on this manual, the TETware GUI, TETware or the TETworks website please email them to [tetware\\_manager@opengroup.org](mailto:tetware_manager@opengroup.org)

All Problem Reports are welcome and actively encouraged. The more problems that are found and fixed the better the product will be. Please submit all bugs and queries found. Also, please submit requests for features and upgrades.

Users should submit problems report using the dedicated email address that is provided for them. Evaluators should email [tetware\\_manager@opengroup.org](mailto:tetware_manager@opengroup.org).



## 17 Exiting the TETware GUI

You can exit the TETware GUI by:

- Using the "Exit" item in the File menu or the Keyboard Shortcut Alt-X.
- Use the system exit

Exiting the TETware GUI is handled differently dependent on the way in which the exit is requested.

**Please note:** It is recommended that you should always exit the TETware GUI via the "File" menu, this is due to the way in which Frame exiting is handled in Java.

The different methods of exiting the TETware GUI are detailed below:

### 17.1 Exiting via the File menu

If you exit the TETware GUI via the File menu then the following checks are made:

#### **Whether there are Test Runs currently executing**

If there any Test Runs that are currently executing then a dialog box will be displayed to confirm whether you wish to stop execution of the Test Runs. The Stop executing test runs dialog box is displayed.

Once you click OK all executing Test Runs will be stopped. This may take some time since the TETware GUI will perform a limited clean up of the Test Runs execution.

#### **If any Test Runs are present in the TETware GUI**

If there any Test Runs open in the TETware GUI you will be prompted to save the Test Runs, saving is not mandatory but is advisable. The Confirm save test runs dialog box is displayed.

If you choose OK then all the Test Runs currently open in the TETware GUI will be saved. If the Test Runs have save files associated with them then they will be saved quietly to those files, if not then the Save As dialog box will be displayed. For more information on saving Test Runs please

see Section 14. If you choose **No** the Test Runs will not be saved and the TETware GUI will exit. The Java Interpreter will also exit.

Once the Test Runs have been stopped (if any are executing) and they have been saved, you will be prompted to exit the TETware GUI with the **Confirm** exit from the TETware GUI dialog box. If you choose **Yes** in this dialog box then the TETware GUI will exit and the Java Interpreter will also exit. If you choose **No** then the TETware GUI will NOT exit.

## **17.2 Using the system exit**

This method of exiting the TETware GUI is dangerous and not recommended if there any Test Runs currently executing. There is no way to halt the TETware GUI closing once a system exit has been called. This means that the TETware GUI is not able to cancel the exit and must try and clean up any Test Runs before exit. The TETware GUI will perform the following checks on exit:

### **Whether there are Test Runs currently executing**

If there any Test Runs that are currently executing then a dialog box will be displayed indicating that any currently executing Test Runs will be stopped. The **Stop test run execution** dialog box is displayed.

Once you click **OK** all executing Test Runs will be stopped. This may take some time since the TETware GUI will perform a limited clean up of the Test Runs execution.

### **If any Test Runs are present in the TETware GUI**

If there any Test Runs open in the TETware GUI you will be prompted to save the Test Runs, saving is not mandatory but is advisable. The **Confirm save test runs** dialog box is displayed.

If you choose **OK** then all the Test Runs currently open in the TETware GUI will be saved. If the Test Runs have save files associated with them then they will be saved quietly to those files, if not then the **Save As** dialog box will be displayed. For more information on saving Test Runs please see Section 14. If you choose either **No** then no Test Runs will be saved and the TETware GUI will exit. The Java Interpreter will also exit.

## Appendix A - TETware License

+++++++TET END USER LICENSE+++++++

BY DOWNLOADING THIS PRODUCT, 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.

TETWARE RELEASE 3 END USER LICENSE  
REDISTRIBUTION NOT PERMITTED  
-----

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

- (A) Free binary evaluation copies - valid for 45 days, full functionality - no warranty,
- (B) Free binary restricted versions - no warranty, limited functionality
- (C) Licensed versions - full functionality, warranty fitness as described in documentation, includes source, binary & annual support.

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

GRANT.

X/Open grants you a non-exclusive license 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 organization which meets X/Open's criteria for a charitable non-profit organization; or (b) your use of the Software is for the purpose of evaluating whether to purchase an ongoing license 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 organizations 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 LICENSE FEES PAID

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

Copyright (c) 1996, 1997, 1998, 1999 X/Open Company Ltd.

Copyright (c) 2000, 2001 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 distributor meeting the following requirements :

- this copyright notice is included unchanged with any binary distribution
- the distributor notifies X/Open
- an annual TET support agreement is 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 derived from other versions of the Test Environment Toolkit, which are copyright

- 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 license.

+++++



## Appendix B - TETware GUI Database Design

### Why did we Create the ‘Database Extract Tool’?

Our objective is to provide TETware users with the ability to store test results in a Standard Database from which they can extract queries and reports providing detailed comparative information relating to test suites, Test Runs and individual test cases.

The extract tool is designed to be compatible with any SQL compliant Database. It extracts journal file parameters and passes them into a Database. Users may then use either the queries or reports provided, or define their own queries and reports using their preferred tools.

The TETware Report Writer facility provides users with a report on a particular Test Run. The Database facility offers the added advantage of allowing users to compare the results from several executions of the same test or Test Run. As such, it provides a powerful test management tool.

It also allows users to add their own database tables so that they may generate reports containing information that is not provided by the TETware GUI.

### What are its Features and Abilities?

The Database extract tool provides users with the means of easily transferring the TETware journal file parameters into most Databases.

It uses standards in order to achieve this:

- Standard Structured Query Language (SQL);
- Open Database Connectivity (ODBC);
- Java<sup>®</sup> Database Connectivity (JDBC).

It is fully supported component of the TETware GUI, so users who encounter difficulties in making their Database connection can use the email support facility to request help from The Open Group Support Team. If you are evaluating TETware please email your questions to [tetware\\_manager@opengroup.org](mailto:tetware_manager@opengroup.org)

## Database Tables

TETware GUI Database comprises three tables. The relationship between the Database tables is shown in Figure 29.

- **configuration\_vars**: This table lists the settings used on Test Runs to specify information about that Test Run, e.g. Configuration Name, Value and File.
- **run\_of\_test\_run**: This table lists the header information that the user requested, e.g. Operators Name, Operating System, Processor type.
- **test\_purpose**: This table lists the detail of the Test Run. This may include information such as the test result, test duration, the time that the test started.

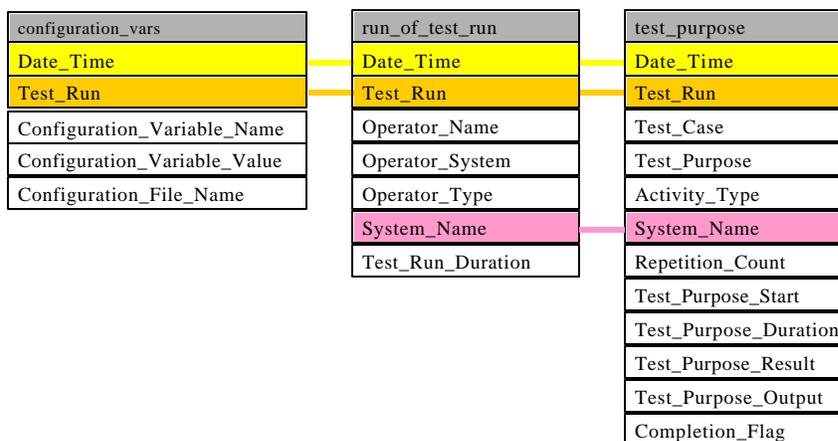


Figure 29: Relationship between database tables

## Database Fields

Within each table, there are a number of fields. In the specimen Database provided these fields have properties that are defined in the tables below:

### run\_of\_test\_run

Field Name	Data Type	Description	Field Size	Format	Decimal Places	Required	Indexed
Date_Time	Date/Time	Defines particular run of a Test Run	n/a	General Date		Yes	Yes (No Duplicates)
Command_Line	Text	Command line passed to controller	255	Normal		No	No
Test_Run	Text	Test Run Name	255	Normal		No	Yes (No Duplicates)
Operator_Name	Text	From First Line 'User:'	255	Normal		No	No
Operating_System	Text	From System Information Line	255	Normal		No	No
Processor_Type	Text	From System Information Line	255	Normal		No	No
System_Name	Text	From System Information Line	255	Normal		No	No
Test_Run_Duration	Number	From Test Case Controller Start and End lines	Long Integer	General Number	0	No	No

**Table 14: run\_of\_test\_run Table**

### configuration\_vars

Field Name	Data Type	Description	Field Size	Format	Decimal Places	Required	Indexed
Date_Time	Date/Time	Defines particular run of a Test Run	n/a	General Date		Yes	Yes (No Duplicates)
Test_Run	Text	Test Run Name	255	Normal		No	Yes (No Duplicates)
Configuration_Variable_Name	Text	Configuration Name	255	Normal		No	No
Configuration_Variable_Value	Text	Configuration Value	255	Normal		No	No
Configuration_File_Name	Text	Configuration File	255	Normal		No	No

**Table 15: configuration\_vars Table**

**test\_purpose**

Field Name	Data Type	Description	Field Size	Format	Decimal Places	Required	Indexed
Date_Time	Date/Time	Defines particular run of a Test Run	n/a	General Date		Yes	Yes (No Duplicates)
Test_Run	Text	Test Run Name	255	Normal		No	Yes (No Duplicates)
Test_Case	Text	Test case name	255	Normal		No	Yes (Duplicates OK)
Test_Purpose	Number	Test purpose number	Integer	General Number	0	No	Yes (Duplicates OK)
Activity_Type	Text	Build/Clean/Exec	5	Normal		No	Yes (Duplicates OK)
System_Name	Text	Name of system participating in the test	255	Normal		No	No
Repetition_Count	Number	Sequence number for repeat directives	Integer	General Number	0	No	No
Test_Purpose_Start	Date/Time	Start time for test purpose	n/a	Long Time		No	No
Test_Purpose_Duration	Number	Duration of test purpose	Integer	General Number	0	No	No
Test_Purpose_Result	Text	Result of test purpose	10	Normal		No	No
Test_Purpose_Output	Text	Output from test purpose	255	Normal		No	No
Test_Purpose_Output_Line	Number	Output Line	Long Integer	General Number	0	Yes	Yes (Duplicates OK)
Completion_Flag	Text	Flag indicating tests that failed to complete	1	Normal		No	No

**Table 16: test\_purpose Table**