Chapter

2

Introduction to the TTCN Suite (in Windows)

This chapter gives an introduction to the TTCN suite and its functionality. It also includes a list of file types supported by the TTCN suite.

When you have read this chapter and want to get more familiar with the TTCN suite, you should read and practice chapter 4, *Tutorial: TTCN Suite Basics (in Windows)*.

If you want to know more about what is new in this release of the TTCN suite, you should read chapter 2, *Release Notes, in the Release Guide*.

> **Note: Windows version**
>
> **This is the Windows version of the chapter. The UNIX version is chapter 3, *Introduction to the TTCN Suite (on UNIX)*.**

Introduction to the TTCN Suite

The TTCN suite is a family of tools for the development, specification and compilation of executable tests for the TTCN language. You can find more information about the TTCN suite tools in "The TTCN Suite Toolset" on page 19.

With the TTCN suite and TTCN, tests are specified in a way that is independent of implementation software and hardware, that is, the tests are only based on the system to be tested and not on the system performing the test. This has encouraged standards organizations to use the technology. Test suites for various standards, supplied by standardization authorities such as ETSI, can therefore be used by all manufacturers and users of that standard. The test suites are created in a graphical editor environment and checked for syntactic and semantic errors by means of a TTCN analyzer.

One of the TTCN suite tools is TTCN Access. It is a general tool for developing translators, code generators for different languages, automatic encoders and decoders, interpreters, reporters and analysis tools for TTCN test specifications.

The TTCN to C compiler produces code for both concurrent and non-concurrent TTCN as well as a limited subset of ASN.1 (see "TTCN ASN.1 BER Encoding/Decoding" on page 56 in chapter 2, *Release Notes*, *in the Release Guide* for further details on the restrictions that apply). To complete the production of an ETS, the generated code must be adapted to the specific target environment.

The generated C code forms an ETS together with the Generic Compiler Interpreter (GCI) Interface. The GCI Interface standardizes the communication between a TTCN component supplied by a vendor and other test system components supplied by the customer, thus forming a Means Of Testing (MOT). GCI is a minimal interface for the adaptation of the generated code to a specific target environment. It makes ETS adaptation easy.

The TTCN Suite Toolset

Telelogic Tau consists of the UML suite. the SDL suite, and the TTCN suite. The TTCN suite is an integrated set of tools used for creating, editing and maintaining TTCN documents. The TTCN suite will be described below. For more information about the SDL suite, see *chapter 2, Introduction to the SDL Suite, in the SDL Suite Getting Started*.

The Organizer – Co-Ordination of Telelogic Tau

You start the TTCN suite by adding and opening TTCN documents in the *Organizer*. The Organizer is the main window in Telelogic Tau. It integrates and co-ordinates the individual tools as required. Several tools may be used simultaneously through the Organizer. For instance, one part of a design may be analyzed at the same time as another is edited.

The Organizer features a graphical view of all the diagrams and documents you are working with, making them an integrated part of the development process. This may include TTCN documents, SDL hierarchies, Messages Sequence Charts, Object Model diagrams, State Charts, Highlevel MSCs and text documents. The view may be freely organized according to your preference.

Different Views of the Test Suite

You can view and edit the contents of a TTCN test suite in the *Browser*, the *Table Editor* and the *Finder*. The Browser gives information about the overall structure while the Table Editor is used for viewing and editing the contents of the TTCN tables. The Finder is used for displaying parts of the test suite, based on different search criteria. Some operations in the TTCN suite produce logs and they are displayed in the *Log Manager*.

The Browser

The Browser presents the view of the overall structure of a test suite. You can manipulate the test suite by adding and deleting items. It is also possible to control the amount of information displayed in the Browser by collapsing and expanding the test suite.

Several Browsers, displaying different test suites, can be opened at the same time. You may also find it useful to create a sub Browser, in which you can select to display only a part of the test suite.

The Table Editor

When you double-click a table that is included in a test suite, it will be opened in the Table Editor. In the Table Editor you can edit the standardized TTCN-GR tables, including the concurrent TTCN tables.

The Finder

By using the Finder, you can display the TTCN tables in a list, without any ordering restrictions. You can sort out the amount of tables included in the list by different criteria, for example name, type, modification time and analysis status. The tables can also be opened and analyzed from the Finder.

The Log Manager

The Log Manager presents information about different operations in the TTCN suite. For example, log information will be generated when you analyze, simulate or export a file.

Building a Test Suite

You can edit the declarations, constraints and dynamic tables manually in the Table Editor, but there are also other ways for editing which reduce the manual work and the risk of typing-errors:

Data Dictionary

An alternative to manually writing behaviour lines is to use the *Data Dictionary*. By using the Data Dictionary, you can select system components – for example PCOs, types, constraints and timers – that are already declared. You can then use these to build behavior statements.

SDL to TTCN Link

SDL to TTCN Link is similar in use and appearance to the Data Dictionary. The difference is that with SDL to TTCN Link, you generate TTCN declarations and interactively build behaviour tables based on an SDL specification.

Autolink

Autolink supports the automatic generation of declarations, constraints and dynamic behaviour tables in a TTCN test suite. The test generation is based on an SDL specification and a number of MSC diagrams. You may create these MSCs manually in the MSC Editor or automatically in

the SDL Simulator or the SDL Validator. The output from Autolink is a test suite in MP file format which can be opened in the TTCN suite and refined subsequently.

In general, it is easier to describe test cases by MSCs than to specify them in TTCN directly. Moreover, Autolink provides a number of facilities to control the format and enhance the readability of a generated test suite.

Generation of the Test Suite Overview

The *test suite overview tables* includes the test suite structure, test case index and default index tables. These tables only need to be generated once, for example when the test suite is to be printed or exported. After that, the overview will be updated automatically as soon as you edit the test suite.

Analyzing, Verifying and Executing a Test

The Analyzer

The *Analyzer* performs a complete syntax check on TTCN and ASN.1 (as used in TTCN), as well as a number of static semantic checks, focusing on the existence and uniqueness of identifiers and the way they are being used.

It is possible to analyze the entire Browser structure or selected parts of it, as well as an entire modular TTCN system. If any tables are found to be erroneous after analysis, the names of those tables will be displayed in the Log Manager. An easy way to find and open an erroneous table, is to left-click the table name in the log and then `<Ctrl>`-right-click the table name again.

The TTCN to C Compiler

The *TTCN to C compiler* translates TTCN to ANSI-C. The TTCN to C compiler reduces the time for code generation as well as the volume of the generated code. Automatically generated code is also free from the potential errors of manual coding. The generated code is independent of the target operative system and protocol. In order to be executable, the generated C code must be linked to a library containing the pertinent OS functions.

The Generic Compiler Interpreter Interface

The *Generic Compiler Interpreter* interface (GCI) is an interface for the adaptation of the generated code to a specific target environment. The GCI interface focuses on what an abstract test suite needs in order to execute in terms of functionality, and on what is needed to integrate TTCN with a larger system.

The TTCN-SDL Co-Simulator

The *TTCN-SDL Co-simulator* allows execution of a TTCN test suite in a host environment. The system under test consists of a simulated SDL system. When the TTCN-SDL Co-simulator is connected with the SDL Simulator, the combined system is simulated. The test cases are executed one by one or in batch. After the execution, the test coverage information can be visualized by the SDL Coverage Viewer.

TTCN Access

TTCN Access makes the internal data structures representing the test suite, available for application programs. It can be seen as a platform for writing applications related to an abstract test suite, for example code generators, interpreters, report tools and analysis tools.

Access provides a default mechanism for accessing the information, but gives you total freedom to override this for one better fitted to your needs.

Input and Output Formats

The TTCN language supports two notations that are equivalent: TTCN-GR – the graphical notation – and TTCN-MP – the textual notation.

TTCN-GR

TTCN-GR is the format used when you edit a test suite in the TTCN suite. The TTCN suite also prints a TTCN document in the TTCN-GR format according to ISO/IEC 9646-3. You may select to print an entire test suite or just parts of it. A preview feature allows you to see what the printed material will look like before printing. The document may be printed as a hard copy or a PostScript file.

TTCN-MP

It is possible to export TTCN documents to TTCN-MP format. Selected items in the Browser – a single table or sets of tables – may also be exported to TTCN-MP format. The test suite that is to be exported, does not necessarily have to be correct, analyzed or complete.

It is also possible to open a document in TTCN-MP format. There is a high tolerance for errors in the MP file and you can use the Analyzer for finding the errors and then correct them in the TTCN suite rather than in the original MP file.

Files in the TTCN Suite

File Types

The TTCN suite supports a number of different file types with various suffixes:

Filename	Explanation
`filename.itex`	TTCN document main data base file
`filename.itex#0`	TTCN document revert file
`filenJAAa002l89,s`	TTCN document working structure file
`filenJAAa002l89,t`	TTCN document working table file
`filename.itex-lock`	TTCN document lock file
`filename.mp`	TTCN-MP file
`filename.log`	TTCN suite log file
`filename.c`	ANSI-C file generated by the TTCN to C compiler
`filename.h`	Header file generated by the TTCN to C compiler

The working *structure* and *table* files are the internal TTCN suite files and will be created temporarily in the temporary directory.The TTCN suite checkpoints automatically to these files during editing but the main data base files are updated only on save.

The revert file is created in the session directory (which is the same as the target directory unless the `ITEX_SESSION_DIR` environment variable is set). It has a name consisting of the name of the main data base file (including extension), a hash sign (#) and a sequence number. The revert file is always stored in the TTCN-GR format even if it has a name like `filename.mp#0`.

Locking of TTCN Documents

When a TTCN document is saved, a lock file for it (which has the additional suffix `-lock`) is automatically created. This means that no other user can access it during the save operation. The lock files are created in the same directory as where the document is saved.

Transferring TTCN Suite TTCN Documents

The entire TTCN suite data bases may be transferred between storage areas. This is useful in order to preserve information that is lost in the TTCN-MP format. Use the usual file commands to achieve this.