NAME

tcc – TETware test case controller

SYNOPSIS

tcc –{**bec**} [*options*] [*test-suite* [*scenario*]]

tcc –{**bec**} **–m** *codelist* [*options*] *old-journal-file* [*test-suite* [*scenario*]]

tcc –{**bec**} **–r** *codelist* [*options*] *old-journal-file* [*test-suite* [*scenario*]]

DESCRIPTION

tcc is the TETware test case controller. It provides support for the building, execution and clean-up of test scenarios.

When TETware-Lite is built, scenarios may only contain test cases which are to be executed on the local system and **tcc** performs all the actions required to process such test cases itself. When Distributed TETware is built, scenarios can contain local, remote and distributed test cases. The distributed version of **tcc** does not perform the actions required to process test cases itself but instead sends requests to the test case controller daemon **tccd** which runs on the local system and also on each participating remote system (see the **tccd**(1) manual page for details).

Apart from the scenario directives which relate to the processing of remote and distributed test cases, the user interface to **tcc** is the same irrespective of whether TETware-Lite or Distributed TETware is being used.

tcc has three modes of operation, namely **build**, **execute** and **clean**, which may be invoked singly or in any combination. These modes are specified by the –**b**, –**e** and –**c** command-line options, at least one of which must appear. All of the other options modify the behaviour of **tcc** in one or more of these operational modes. Each mode (with optionally modified behaviour) is applied to the test cases and invocable components selected for processing.

By default, **tcc** builds, executes or cleans test cases in the named *scenario* contained in the scenario file **tet_scen**, which is located in the test suite root directory for *test-suite* (see DIRECTORIES below). If no *scenario* is specified, the default scenario named **all** is used. If no *test-suite* is specified, **tcc** attempts to deduce a default test suite name using the following rules:

1. If the **TET_SUITE_ROOT** environment variable is set and the current directory lies under the directory hierarchy specified by this variable, then the test suite is the component of the current directory's path name which lies immediately below **$TET_SUITE_ROOT**. For example, if **$TET_SUITE_ROOT** is **/usr/tet3** and the current directory is **/usr/tet3/suite1/results**, then the name of the default test suite is **suite1**.

2. If the **TET_SUITE_ROOT** environment variable is not set and the current directory lies under the directory hierarchy specified by the **TET_ROOT** environment variable, then the test suite is the component of the current directory's path name which lies immediately below **$TET_ROOT**.

3. If the current directory lies outside of the directory hierarchy specified by the **TET_SUITE_ROOT** environment variable (if set) or the **TET_ROOT** environment variable (if **TET_SUITE_ROOT** is not set), then no default test suite name can be deduced.

DIRECTORIES

By default, **tcc** interprets test case names relative to the **test suite root** directory. The location of this directory is determined as follows on the local system:

1. If the **TET_SUITE_ROOT** environment variable is set, the **test suite root** directory is determined by the test suite name, relative to **$TET_SUITE_ROOT**.

2. If the **TET_SUITE_ROOT** environment variable is not set, the **test suite root** directory is determined by the test suite name, relative to **$TET_ROOT**.

3. If the **TET_RUN** environment variable is set, then the directory subtree below the **test suite root** (determined as described above) is copied to the location below **$TET_RUN** and this location becomes the new **test suite root** directory.

However, an alternate execution directory on the master system may be specified by the **TET_EXECUTE** environment variable or by a command-line option (see OPTIONS below). If an alternate execution directory is specified, **tcc** interprets test case names relative to this directory when operating in execute mode.

By default, **tcc** creates a directory called **tet_tmp_dir** below the test suite root directory. However, a different temporary directory name on the local system may be specified by the **TET_TMP_DIR** environment variable. Each invocation of **tcc** creates a unique subdirectory below the temporary directory on startup and removes it and its contents on normal completion.

CONFIGURATION FILES

During execution, **tcc** reads configuration variables from certain configuration files on both the local and the remote systems (if any). By default, the name of the build mode configuration file is **tetbuild.cfg**, that of the execute mode configuration file is **tetexec.cfg** and that of the clean mode configuration file is **tetclean.cfg**. The build and clean mode configuration files reside in the test suite root directory on each system. The execute mode configuration file resides in the alternate execution directory if one has been specified, otherwise in the test suite root directory.

Variables used in all three modes of operation that relate to remote and distributed testing are read from the file named **tetdist.cfg** in the test suite root directory on the local system. This file must at least contain assignments for the **tet root** and test suite root directories for any remote systems that are specified in the scenario being processed.

JOURNAL FILE

By default, **tcc** creates a sequentially numbered directory below the **results** directory in the test suite root directory for the named *test-suite* on the local system, and places the journal file and saved intermediate result files there. On startup, **tcc** writes the name of the journal file being used to the standard output.

RESULT CODES

tcc uses a table of result codes to interpret the results generated by API-conforming test cases. A default table containing standard codes is built in to **tcc**. It is possible to specify additional codes in user-supplied result codes files located below the **tet root** and test suite root directories on the local system. These files are optional but, if they exist, the codes specified in them are added to the table of standard codes. The default name for each of these files is **tet_code** but this name can be changed by means of the TET_RESCODES_FILE configuration variable.

OPTIONS

The following *options* alter the default behaviour described above:

–a *directory*
Use *directory* as the alternate execution directory instead of the one specified by the **TET_EXECUTE** environment variable (if any).

–f *file* Use *file* as the clean mode configuration file instead of the default.

–g *file* Use *file* as the build mode configuration file instead of the default.

–i *directory*
Place the default journal file and saved intermediate results files in *directory* instead of in the default location.

–j *file* Use *file* as the journal file instead of the default.

−l *scenario-line*
: Process *scenario-line* as if it appeared in a scenario file below a scenario named **all**. More than one **−l** option may be specified; the *scenario-lines* are processed in the order in which they appear on the command line. *scenario-line* must be presented as a single argument so it must be quoted if it contains embedded spaces. If a scenario file is specified by a **−s** option, any *scenario-lines* are processed before that scenario file is read. If no **−s** option is specified, the default scenario file **tet_scen** is not read when **−l** is used.

−n *string*
: Do not process test case names that contain *string*. More than one **−n** option may appear.

−p Enable progress reporting. As each build, execute or clean operation is started, a line indicating the time, mode and scenario line being processed is printed on the standard output.

−s *file* Use *file* as the scenario file instead of the default.

−t *timeout*
: Terminate the build, execute or clean of an individual test case if processing would continue for more than *timeout* seconds.

−v *variable=value*
: The specified configuration *variable* is set to *value*, overriding any assignment in the configuration file for the current mode. It is probably best to surround *value* with single quotes if it contains characters which have special meaning to the Shell. More than one **−v** option may appear.

−x *file* Use *file* as the execute mode configuration file instead of the default.

−y *string*
: Only process test case names that contain *string*. More than one **−y** option may appear. The **−n** option has higher precedence than the **−y** option; thus, a test case is not processed if its name is matched by *strings* specified with both the **−n** and the **−y** options.

RERUN AND RESUME OPTIONS

The following options are mutually exclusive:

−m *code-list*
: Causes **tcc** to resume the previous run of the specified *scenario* in the named *test-suite* whose results are in *old-journal-file*. *code-list* specifies the point in the previous run from which processing is to be resumed and may consist of a comma-separated list of result codes, or of one or more of the letters **b**, **e** and **c** to specify failures in particular processing modes. If *code-list* consists of result codes, then processing resumes at the first invocable component whose result in the previous run matched one of those in the list. If *code-list* specifies processing modes, then processing resumes at the first test case which failed to build or clean or the first invocable component which, when executed, did not report PASS in the previous run.

For example:

tcc −b −m b

Resume building from the first test case that failed to build.

tcc −e −m FAIL,UNRESOLVED

Resume execution from the first invocable component that reported FAIL or UNRESOLVED.

tcc −bec −m b,e

Resume building, execution and cleaning from the first test case which failed to build or from the first invocable component that did not report PASS.

−r *code-list*

Causes **tcc** to re-run individual test cases and invocable components from the specified *scenario* in the named *test-suite* whose results are in *old-journal-file*. *code-list* specifies the elements that are to be re-run and may consist of a comma-separated list of result codes, or of one or more of the letters **b**, **e** and **c** to specify failures in particular processing modes. If *code-list* consists of result codes, then test cases and invocable components are re-run if the corresponding result in the previous run matched one of the result codes in the list. If *code-list* specifies processing modes, then a test case is re-run if it failed to build or clean and an invocable component is re-run if it did not report PASS when it was executed in the previous run.

For example:

tcc −b −r b

Re-build test cases that previously failed to build.

tcc −e −r FAIL,UNRESOLVED

Re-execute all invocable components that previously reported FAIL or UNRESOLVED.

tcc −bec −r b,e

Re-build, execute and clean all test cases that previously failed to build or execute, and all invocable components that did not previously report PASS when executed.