

## *The TTCN Browser (on UNIX)*

This chapter contains a reference manual to the TTCN Browser. The Browser presents an overview of the TTCN document. In the Browser, it is possible to edit the TTCN document structure, to determine which parts of the document to view, and to apply different tools.

The functionality of the Browser, the menus, windows and quick buttons, are described in this chapter. The TTCN Table Editor is described in chapter 26, *The TTCN Table Editor (on UNIX)*. How to analyze and find tables is described in chapter 27, *Analyzing TTCN Documents (on UNIX)*.

For an overview of the TTCN suite, see chapter 3, *Introduction to the TTCN Suite (on UNIX), in the TTCN Suite Getting Started*.

### **Note: UNIX version**

This is the UNIX version of the chapter. The corresponding Windows chapter is chapter 31, *Editing TTCN Documents (in Windows)*.

## Introduction to the TTCN Browser

The TTCN Browser is the part of the TTCN suite that presents TTCN document structures. When a new TTCN document is opened, place holders for the static parts of the document are already created. Then you can add and delete the dynamic parts.

The amount of information actually displayed in a Browser can be collapsed and expanded. It is also possible to display selected parts of the TTCN document in a sub Browser. You open a sub Browser from the main Browser or from another sub Browser.

The functionality of a menu choice is applied to all items or selections of items in the Browser. If an operation is invoked from the Organizer (see [chapter 2, \*The Organizer\*](#)), it is always applied to all items in the TTCN document.

### Opening the Browser and a TTCN Document

You open the TTCN Browser by selecting a TTCN document already included in the Organizer followed by the menu choice *Edit*, or by double-clicking the TTCN document. You may also add a new TTCN document with *Add New* or *Add Existing* from the *Edit* menu, or start a Browser directly by selecting *Editors > TTCN Browser* from the *Tools* menu. See [chapter 2, \*The Organizer\*](#) for more information.

## The Browser User Interface

For a general description of the Telelogic Tau user interface, see [chapter 1, \*User Interface and Basic Operations\*](#).

The TTCN document in the Browser contains *items*. These are:

- *Static items*, common to every TTCN document, such as the declarations part or constraints part.
- *Dynamic items* – that is TTCN objects – such as tables, groups or objects in a table, for example test case variables.

The Browser displays all the static items in the order defined in the TTCN standard. This ordering cannot be changed. For example, it is not possible to have the constraints part coming before the declarations part. Only the dynamic items – the tables – may be added and deleted. The structure is indicated by indentation.

# The Browser User Interface

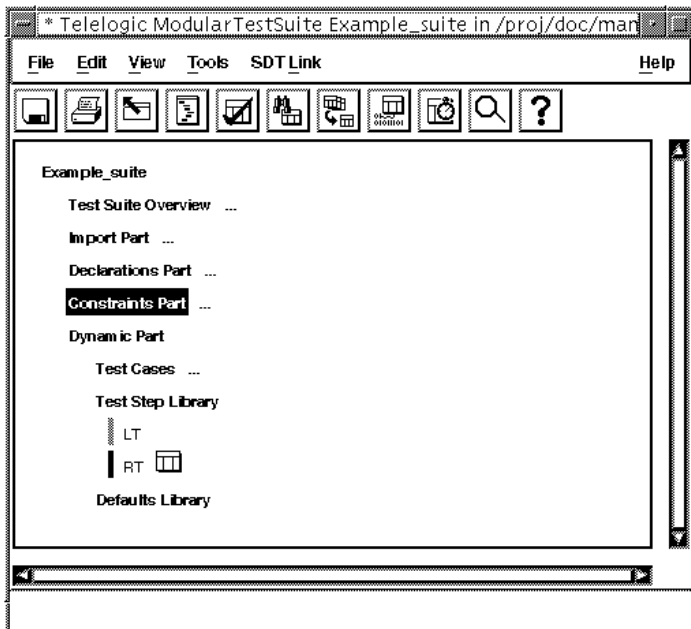


Figure 187: The the TTCN Browser window on UNIX

Figure 187 depicts the Browser window. The test suite overview, the import part, the declarations part, the constraints part, the test cases and the defaults library are collapsed. This is indicated by the three dots.

The dynamic part and the test step library are expanded. The test step library contains two test steps: LT and UT. LT has been edited but not analyzed. This is indicated by the shaded bar (gray on a color screen). The table icon indicates that UT is opened and the filled bar (red on a color screen) that UT has been analyzed and is erroneous. Items with no analysis bars have been analyzed and contain no errors.

Items in **bold** font represent the static (or structural) parts of the TTCN document and cannot be opened or edited. The items in normal font represent named TTCN tables which can be opened and edited.

Some of the static items may remain unused. For example, if *Test Case Variables* is not used it will still be displayed in the Browser. However, it will not be printed or output in the TTCN-MP format.

## Controlling the Items in the Browser

### Selecting Items

There is a difference between *selection* and *input focus*. Selection is an inverted item or text string. The input focus is indicated by a rectangular border surrounding the relevant item. Keyboard commands operate on the item with input focus, not the selection.

Point and click once to select a **single** item in a Browser – note that this also sets the input focus. Selecting a new item will deselect the previous item (if any).

To select **several** items, press `<Ctrl>` while selecting items. Note that the input focus is set to the last selected item. Pressing `<Ctrl>` while selecting items works in toggle mode: a selected item will be deselected and vice versa.

When clicking the **middle** mouse button, the input focus will be set to the new item. The selection will not be changed.

You can also select items by using:

- The popup menus, see [“Using Popup Menus” on page 1116](#).
- Selector, see [“Using More Complex Selections” on page 1117](#).
- Compare, see [“Comparing TTCN Documents” on page 1130](#).

#### Hint:

Some of the commands in the TTCN suite only work when a single item is selected in a Browser. Before applying such commands, it is recommended that you deselect all items before you select a single item.

## Controlling What is Displayed

The *View* menu is used for deciding how much of the test suite that should be displayed in a Browser. This is useful when you work with large test suites. This menu also contains commands to access any opened Table Editor and to close all opened editors, as well as to access any loaded TTCN document.

### ***View > Collapse***

Hides all items under the marked item. A collapsed item is indicated by an ellipsis (...). A shortcut for *Collapse* is to type <c> without invoking the *View* menu.

### ***View > Expand***

Displays all items at the next level below the marked item. A short-cut for *Expand* is to type <e> without invoking the *View* menu.

### ***View > Expand Tree***

Displays all items at all levels below the marked item. A short-cut for *Expand Tree* is to type <Shift+e> without invoking the *View* menu.

### ***View > Close All Editors***

Closes all open Table Editors, regardless of which documents they belong to.

### ***View > Show Editor***

This sub menu presents a list of all open Table Editors, in any TTCN document. By selecting an item in this list, the corresponding Table Editor is brought forward.

### ***View > Show Document***

This sub menu presents a list of all loaded TTCN documents. By selecting an item in this list, the main Browser of the corresponding document is brought forward.

## Creating a Sub Browser

Sub Browsers may be created to display selected portions of a test suite. These selections may be made manually or with the Selector (see above).

This facility is useful when working with a large test suite or with items that fulfil certain criteria.

### ***Tools > Browser***



Causes a sub Browser to be displayed containing only the selected items from this Browser. There is no limit on the number of sub Browsers that may be displayed.

#### **Note:**

Closing the main Browser also closes any sub Browsers that are opened.

## Editing the TTCN Document Structure

The TTCN suite displays all the components of a TTCN document in the structured format of the Browser. Part of this structure is static and common to every TTCN document. This static structure is automatically generated when the TTCN document is first created.

To build an individual TTCN document, named tables (such as PDUs, Constraints and Behaviours) and named objects to the multi-object tables (such as Test Case Variables, PCOs and Timers) must be added to this static structure.

The *Edit* menu is used to add, delete, copy, cut, paste etc. *editable items* (i.e. groups, tables or objects in a multiple object table) in a Browser.

## Adding and Inserting Items

### ***Edit > Add***

Adds a new table to the Browser or adds an object to a table in the Browser.

In the case of single-object tables, select the Browser item that “holds” the table. Choosing *Add* will add a new table to the end of the list of tables. The new table will be unnamed. Use the *Rename* command to give the table its correct name, or open the table and edit the name there.

### **Example 185**

---

To add a new TTCN PDU to a Browser, select the item *TTCN PDU Type Definitions* and apply *Add*.

---

In the case of multiple-object tables, select the item that is the name of the table. Choosing *Add* will add a new object to the end of the list of objects in this table. The new object will be unnamed. Use the *Rename* command to give the table its correct name, or open the table and edit the name there.

### **Example 186**

---

To add a new Test Case Variable to a Browser, select the item *Test Case Variables* item and apply *Add*.

---

### ***Edit > Insert Before***

Similar to *Add* but the new table (or object in a table) is inserted **before** a selected table (or selected object).

#### **Note:**

Only a single item should be selected.

### ***Edit > Insert After***

Similar to *Add* but the new table (or object in a table) is inserted **after** a selected table (or selected object).

#### **Note:**

Only a single item should be selected.

## Adding and Inserting Groups

The following commands are used to add/insert new groups to the Test Case, Test Step and Default libraries.

### ***Edit > Add Group***

Adds a new Test (Case) Group, a new Test Step Group or a new Default Group in the Dynamic Part. The new group is added at the **next level after** the selected group.

Use the *Rename* command to edit the temporary name *NoName*. Note that the trailing slash (/) is automatically added.

To add a table to a group, select the group and use the *Add* command.

### ***Edit > Insert Group Before***

Inserts a new group **before** a selected item.

### ***Edit > Insert Group After***

Inserts a new group **after** a selected item.

## Adding and Inserting Compact Tables

It is possible to specify that all the Test Cases in a given group are displayed in the compact format (see Annex C, clause C.3 of ISO/IEC 9646-3). The following commands allow the insertion of compact groups in the test suite hierarchy:

### ***Edit > Add Compact Group***

Adds a compact group at the **next level after** the selected group.

### ***Edit > Insert Compact Group Before***

Inserts a compact group before a selected item.

### ***Edit > Insert Compact Group After***

Inserts a compact group after a selected item.



## Sorting Items

The places where the *Add* command may be used, may also be sorted.

### ***Edit > Sort***

Sort the list of items below the selected item lexicographically.

## Cutting, Copying and Pasting Items

Editable items may be cut or copied to the clipboard. The contents of the clipboard may be pasted according to compatible classes, for example:

- Test Suite Parameters, Test Suite Constants, Test Suite Variables and Test Case Variables may be pasted into each other.
- TTCN ASP, PDU, Structured Type and CM Definitions may be pasted into each other and into TTCN ASP, PDU, Structured Type and CM Constraints (and vice versa).
- ASN.1 ASP, PDU, Structured Type and CM Definitions may be pasted into each other and into ASN.1 ASP, PDU, Structured Type and CM Constraints (and vice versa).
- Test Case, Test Step and Default Behaviours may be pasted into each other.

However, it is not possible to paste a Constraint for example as a Test Case Behaviour.

The *Paste* command is not available if the clipboard contains an object of an incompatible type to the selected object.

### **Note:**

On the table row and text levels, no paste restrictions apply.

### ***Edit > Cut***

Deletes the selected items from the Browser and stores them in the clipboard.

### ***Edit > Copy***

Copies selected items from the Browser to the clipboard.

**Edit > Paste Before**

Pastes the contents of the clipboard before a selected item.

**Edit > Paste After**

Pastes the contents of the clipboard after a selected item.

**Edit > Paste In**

Pastes the contents of the clipboard to the end of a list of tables (or list of objects in a multi-object table).

**Deleting Items**

Editable items may be removed from the Browser.

**Edit > Delete**

Deletes selected items from the Browser.

**Caution!**

The TTCN suite has no undo so this command is irreversible.

**Browser Shortcut Keys for Navigation**

The cursor and mouse may be used to set the input focus on an item in a Browser. However, you may find the following shortcut keys more useful:

- `<Down Arrow>` moves the input focus to the **next** item. This movement is not sensitive to the structure of a Browser.
- `<Shift+Down Arrow>` moves the input focus to the next item at the **same** level of indentation.
- `<Up Arrow>` moves the input focus to the **previous** item. This movement is not sensitive to the structure of a Browser.
- `<Shift+Up Arrow>` moves the input focus to the **previous** item at the **same** level of indentation.
- `<Left Arrow>` moves the input focus to the **previous** level of indentation.

## Editing the TTCN Document Structure

---

- `<Shift+Left Arrow>` moves the input focus to the **previous** level of indentation and collapses that branch in a Browser.
- `<Right>` moves the input focus to the **first** item at the **next** level of indentation. If the next level is collapsed then it is automatically expanded.
- `<Shift+Right Arrow>` moves the input focus to the **last** item at the **next** level of indentation. If the next level is collapsed then it is automatically expanded.

### Renaming Dynamic Items

The *Rename* command is used to give names to newly created Browser items that have the temporary name *NoName* or to rename existing Browser items. It is available in the popup menu for the item.

#### **Rename**

Renames the dynamic item. A text box with a cursor appears for the item. Type the new name and then press `<Return>`. It is also possible to copy, cut, paste and delete text.

- `<Copy>` or `<L6>` on a Sun keyboard copies selected text.
- `<Cut>` or `<L10>` on a Sun keyboard cuts selected text. `<Ctrl+k>` cuts text from the current position to the end of the line.
- `<Paste>` or `<L8>` on a Sun keyboard pastes text that is stored in the paste buffer. `<Ctrl+y>` pastes the most recently cut line at the current position.
- `<Delete>` or `<Backspace>` deletes one character to the left of the insertion point. `<Ctrl+d>` deletes one character to the right of the insertion point.

#### **Note:**

The TTCN suite has no undo function. This means that `<Undo>` has no effect.

A shortcut for *Rename* is to type `<r>` without invoking the pop-up menu. The item with the input focus will then be renamed.

## Opening a Table

A table is opened from a Browser when you double click on the specific item. Observe that this command has the additional effect of deselecting every item in the Browser. If this additional effect is undesirable, it can be inhibited by holding down `<Ctrl>` simultaneously.

In the case of single-object tables, point to the table name (e.g. Test Case name). To open a multiple-object table, point to a Browser item that is the title of the table (e.g. Test Case Variables) or to an object in the table (e.g. a Test Case Variable name).

In the case of collapsed items, point and double-click to expand the item (e.g. displaying all items at the next level below the pointed item). A double-click on a group table (regardless if it is collapsed or not) opens the group table. If the item pointed at is a leaf node that is not a table, the double-click will instead create a new item below it.

Alternatively, you may point and click once with the middle mouse button (i.e. set the input focus) and then press `<t>` to open the table. Note that the input focus can also be set with shortcut keys. See [“Key and Button Bindings”](#) on page 1155.

If a table has previously been opened, but it is for some reason hard to get at, it may be brought forward by selecting the name of the table in the submenu *Show Editor* of the menu *View*.

## Printing a TTCN Document

To print a TTCN document:



- Select *Print* from the *File* menu.
  - The shortcut is `<Ctrl+P>`.

For more information, see [“The Print Dialogs in the TTCN Suite”](#) on page 324 in chapter 5, *Printing Documents and Diagrams*.

# Requesting License Information

### **Help > License Info**

The TTCN suite consists of several tools and for running each tool it is required that the appropriate license is installed. The the TTCN suite *License Information* dialog is used to list these licenses.

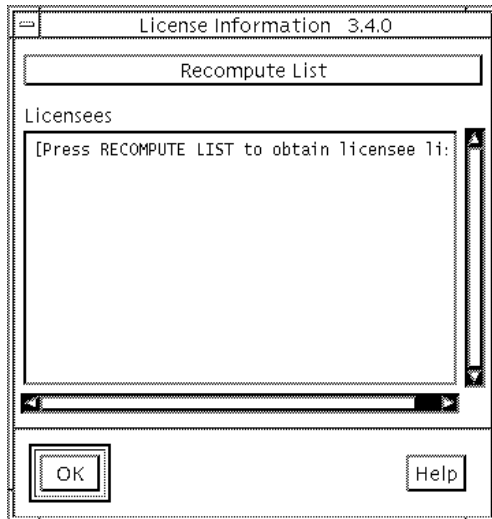


Figure 188: The Information dialog

- Clicking *Recompute List* retrieves the list of licensees and lists them in the scrollable area below.
- The *Licensees* area shows a list of licensees and the licenses available.

## Using Popup Menus

### Node Specific Popup Menu

This menu is shown when you point to an item in the Browser and click the right mouse button. Most of the commands are also available in the main menu bar. The commands in this menu are applied on the pointed item. Using this menu moves the input focus to the item pointed to, but does not change the selection.

### Background Popup Menu

This menu is shown when you point to the background of the Browser and click the right mouse button. Using this menu does not affect the selection or the input focus.

### Popup Menu Commands

The following commands in the popup menus are not available elsewhere.

#### ***Select All***

Selects all items in the Browser. The same effect can be achieved by pressing `<Ctrl+>`.

#### ***Deselect All***

Deselects all selected items. The same effect can be achieved by pressing `<Ctrl+\>`.

#### ***Toggle Select***

Toggles the selection state for the item. The same effect can be achieved by pressing `<Ctrl+space>`.

#### ***Toggle Select Tree***

Toggles the selection state for the whole sub-tree below the item. The same effect can be achieved by pressing `<Ctrl+Shift+space>`.

#### ***Rename***

Renames the item, see [“Rename” on page 1113](#).

# Using More Complex Selections

Apart from manually selecting items in the Browser, you can make more complex selections by using the Selector. These selections can then be displayed in a sub Browser.

The Selector tool is available from the Browser *Tools* menu. Like the other TTCN suite tools, the Selector tool works on selections. The purpose of the Selector tool is to, from an existing selection, create a new selection.

With the Selector, you can make selections in three ways:

- By restricting the existing selection
- By making additions to the existing selection
- By replacing the existing selection

You can make selections by specifying restrictions on table names, restrictions on tables types and even restrictions on the contents of the tables.

Restrictions can also include the analysis status of the table, i.e. *not analyzed*, *error* and *OK*.

Selections can also be made depending on references between TTCN objects. A typical use of this feature might be to select (and perhaps later Convert to TTCN-MP) an entire Test Case, i.e. the Test Case Behaviour and all its associated declarations, constraints, Test Steps etc.

### **Note: UNIX only**

The Selector is only available on UNIX.

### ***Tools > Selector***

Creates a new selection from a given selection. The Selector dialog is opened:

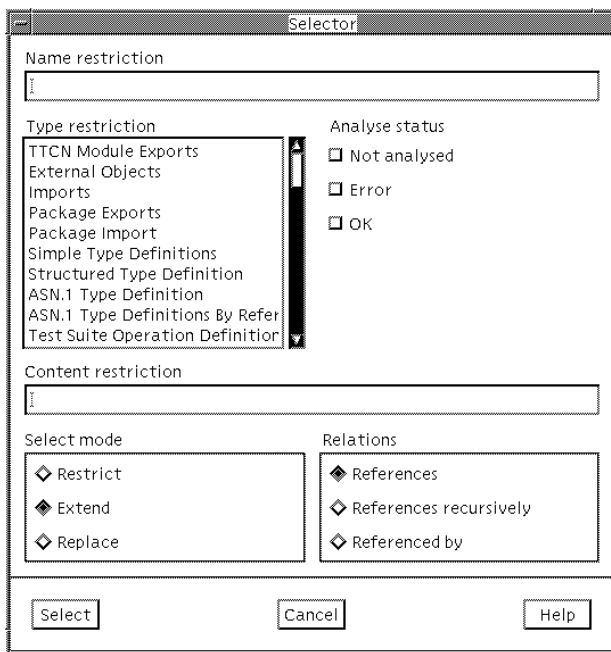


Figure 189: Selector dialog

### Name Restriction

Enables the user to define the restrictions on the **identifiers** of the named TTCN tables (e.g. ASPs, PDUs, Constraints, Test Cases, Test Steps and Defaults) to be selected. This restriction is a regular expression (see “[Regular Expressions](#)” on page 1123). If the name restriction field is empty then no restrictions on the table names will apply.

### Example 187

The name restriction `^abc` will select all tables whose names begin with the string “abc”.

### Type Restriction

Enables the user to define the restrictions on which **kind** of TTCN tables that are to be selected. Click once to select one or more table types.



## Using More Complex Selections

---

### Example 188

---

Suppose that the initial selection is the entire Dynamic Part. Choosing the table types Test Case Dynamic Behaviour and Test Step Dynamic Behaviour **together** with the name restriction  $\wedge abc$  will select all test cases and test steps whose names begin with the string “abc”.

---

If no table types at all are selected then no type restrictions are assumed to apply (i.e. the same effect as selecting all table types).

### Content Restriction

Enables the user to define the restrictions on the **contents** of TTCN tables that are to be selected. This restriction is a regular expression (see “[Regular Expressions](#)” on page 1123). If the content restriction field is empty then no restrictions on the contents of tables will apply.

### Example 189

---

Suppose that the initial selection is the entire TTCN document. Choosing the table type TTCN PDU Type Definition **together** with the Content restriction  $\wedge xyz$  will select all TTCN PDUs which have a field beginning with the string “xyz”.

### Select Mode

Enables the user to choose a select mode: *restrict*, *extend* or *replace*.

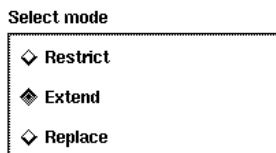


Figure 190: Selector mode

- In the *Restrict* mode the initial selection will be **reduced** to those items which fulfill the user supplied restrictions. The examples above illustrate the *Restrict* mode.

Note that in this mode the *Relations* option is not available.

- In the *Extend* mode the initial selection will be **extended** with the items that 1) fulfill the references given by the *Relations* option and 2) fulfill the user supplied restrictions (if any).
- In the *Replace* mode items in the initial selection will be **replaced** by the items that 1) fulfill the references given by the *Relations* option and 2) fulfill the user supplied restrictions (if any).

### **Relations**

When a TTCN document is analyzed, the TTCN suite builds a symbol table of all the named TTCN objects that are in the TTCN document. If required, the Selector will perform a ‘look-up’ in this table in order to select referenced objects.

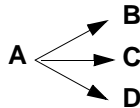
It is required that the TTCN document is analyzed before applying the Selector with the *Relations* option. This will ensure that the entries in the symbol table are consistent with the TTCN document on the screen.

There are three kinds of references: *References*, *References recursively* and *Referenced by*

- *References*

Selects all other objects that are referenced by the initially selected objects. This new selection may be further restricted if any *Name* and/or *Type* and/or *Content* restrictions are chosen.

In the first case, suppose that we have object A. Choosing *References* will select all the other objects, e.g. B, C and D that object A references.



---

### **Example 190**

If the initial selection is a single test case and we make a type restriction of Test Step Dynamic Behaviour but no Name or Content restrictions, then choosing *Extend* and *References* will add to the current selection all the test steps (if any) that the selected test case references.

---

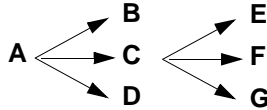
## Using More Complex Selections

---

- *References recursively*

Same as above except that all recursive references are also selected.

The second case is the same as the first, except that the references are selected recursively, e.g. B, C, D, E, F and G.



---

### Example 191

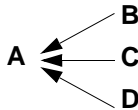
To extract an entire test, select the desired test case and apply *Extend* and *References recursively* without any other restrictions. The Selector will select the Test Case Behaviour and all its associated declarations, constraints, attached Test Steps (including any Test Steps that these Test Steps may attach), Defaults etc. The *Convert to MP* command can then be used to output this test case in TTCN-MP format.

---

- *Referenced by*

Selects all objects that have references to the selected object. This new selection may be further restricted if any *Name* and/or *Type* and/or *Content* restrictions are chosen.

In the third case suppose that we have object A. Choosing *Referenced by* will select all the other objects, e.g. B, C and D that reference object A.



---

### Example 192

If the initial selection is a single PDU definition, then choosing *Extend* and *Referenced by* without any other restrictions will select all the objects (e.g. constraints and behaviours) that use this PDU type.

---

## Keeping and Retrieving a Selection

### *File > Keep Selection*

Saves the current selection for later retrieval. This operation may, for example, be used when an Access application is to be applied on the TTCN document. To be able to retrieve the saved selection in another session the *Save Document* command needs to be issued after this command. For more information see [“Getting Started with TTCN Access” on page 978 in chapter 22, \*TTCN Access\*](#).

### *File > Retrieve Selection*

Retrieves the saved selection.

## Searching by Using the Selector

A search pattern (regular expression) can be defined for either the complete content of a set of objects or only for the names.

A search always operates with a Browser selection as its scope. It will produce another selection as the result. By creating a sub Browser from the resulting selection a convenient environment for viewing the objects that fulfilled the search predicate is achieved.

## Making Sub Browsers from Selections

Often, after making a selection, it is useful to make a sub Browser which only contains the items which are selected. This can easily be done by using the *Browser* command from the *Tools* menu of the Browser.

### **Example 193** ---

Applying the Selector on an entire TTCN document with the *Analyze status* option set to *error* will select all objects that have syntax and/or static semantic errors. Choosing the *Browser* command from the *Tools* menu of the Browser where the selection was applied will create a Sub-browser that only contains the error objects (including the structure needed to “glue” them together).

---

### Regular Expressions

When you use the Selector or search from the Table Editor, string patterns are used for matching and searching for named objects or text strings. In UNIX terminology these patterns are called *regular expressions*. Regular expressions include symbols specifying the pattern to be searched for. Some of these symbols are:

\ < > ^ \$ [ ] \*

Use of these symbols in regular expressions involves a quite complicated syntax which can be very powerful. We illustrate some common uses below, but for full information we suggest that the user takes a look at the UNIX manual entry for *regexp* (do a *man regexp*), and look under the heading Regular Expressions.

#### Example 194

---

TTCN tables contain *fields*. Fields contain text *strings*. Text strings contain *words*.

The pattern `abc` will match all words that **include** the string “abc”, e.g. `abcxyz xyzabc, xyzabcxyz, abc`.

The pattern `\<abc\>` will match all **words that are exactly** “abc”. This is useful for finding TTCN identifiers embedded in strings and comments.

The pattern `^abc` will match any field that **begins** with the string “abc”, e.g. `abcxyz abc, abcabc`.

The pattern `abc$` will match any field that **ends** with the string “abc”, e.g. `xyzabc, abc, abcabc`.

The pattern `^abc$` will only match **fields containing exactly** “abc”.

The pattern `abc[1-5]$` will select all fields that end with “abc” and a single number between 1 and 5, e.g. `abc1` and `xyzabc4`, but not `abc7` or `abc12`.

---

## Searching and Replacing

You specify the text to search for and the replacement text in the Table Editor, see [“Searching and Replacing” on page 1168 in chapter 26, \*The TTCN Table Editor \(on UNIX\)\*](#). However, you can invoke the search from the Browser too and in this case, an empty Table Editor will be opened.

### ***Edit > Find & Replace***



Opens an empty Table Editor where you can specify what to search for.

# Presenting Status Information

The Reporter presents status information on selected Browser items (i.e. TTCN objects) in the form of a simple list. This list can include such information as analyze status and modification dates etc. The Reporter also provides limited formatting of the list.

Reporter output may be stored to file for later use, for example printing or as input to an AWK script for further processing.

### Note: UNIX only

The Reporter tool is only available on UNIX.

## Constructing Lists with the Reporter

The Reporter tool is available from the Browser *Tools* menu.

### *Tools > Reporter*

The Reporter tool is applied to selected items in a Browser, and presents them in the form of a simple list. Selections can either be made by using the mouse or by using the Selector tool (see [“Using More Complex Selections”](#) on page 1117).

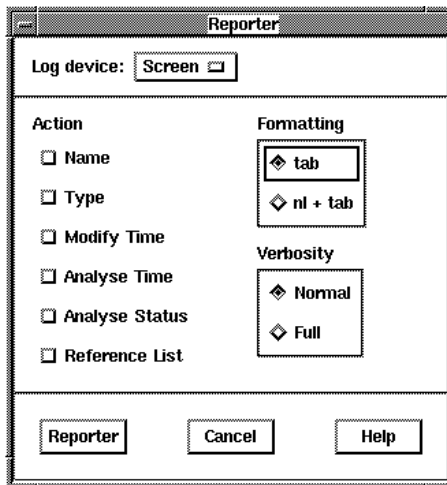


Figure 191: Reporter dialog

**Log Device**

Controls the log (output) device for the Reporter tool. The default value is *Screen*, but the log can be directed to a named file by choosing *File* or turn it off altogether by choosing *None*. The file extension for the reporter log is `.rpt`.

For a full description of this command see “The TTCN Suite Logs” on page 6 in chapter 1, *User Interface and Basic Operations*.

**Action**

The following options specify which information about each selected item shall appear in each Reporter list entry.

- *Name*  
The name of the item (e.g. ESTABLISH\_CONNECTION)
- *Type*  
The type of the item (e.g. Test Step Dynamic Behaviour)
- *Modify Time*  
The date and time that the item was last modified (edited) (e.g. Thu Aug 20 13:05:55 1992).
- *Analyze Time*  
The date and time that the item was last analyzed (e.g. Thu Aug 20 13:08:35 1992).
- *Analyze Status*  
The analyze status of the item (e.g. error)
- *Reference List*  
The other objects that this object references. This information is used, for example, by the Selector when the References option is set (e.g. N\_SAP:L:LT\_DEFAULT:CR1:NDR:CR:CC1:NDi:CC).



# Presenting Status Information

---

## **Formatting**

The list can be formatted using either *tab* or *nl + tab*, where *nl* stands for *new line*.

- *tab*

In this format the components of each list entry are separated by a tab.

### **Example 195**

---

Part of a Reporter log showing two test steps. The list was generated using only the *Name* and *Type* option with the format option set to *tab*.

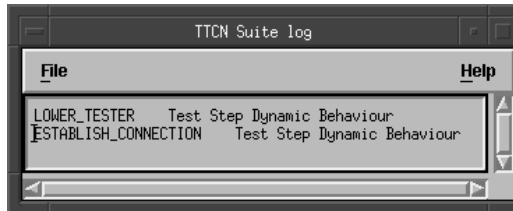


Figure 192: The TTCN suite log for Reporter (*tab*)

- *nl + tab*

In this format the components of each list entry each are printed on a new line. Each component, except the first component in the list entry, is preceded by a tab.

**Example 196**

Part of a Reporter log showing two test steps. The list was generated using all the options and with the *nl + tab* format.

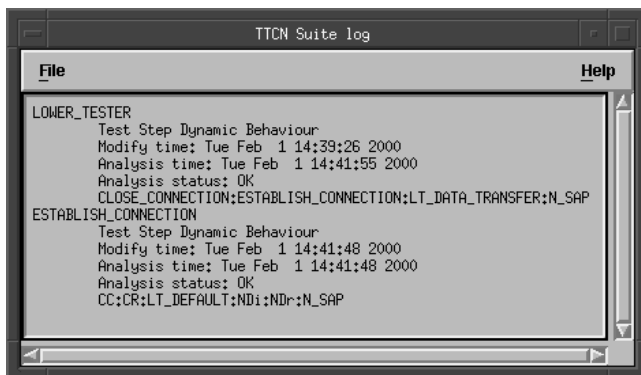


Figure 193: The TTCN suite log for Reporter (*nl+tab*)

**Verbosity**

Full verbosity gives more detailed information. The following extra information is available with full verbosity:

- *Reanalysis*

It is obvious that if the item is not OK (its analyze status), it needs to be reanalyzed. Even if the item has an OK status, the item may need to be reanalyzed since it may contain references to items which have been modified. The reporter with full verbosity will determine if the item need to be reanalyzed or not. This information is reported together with the analysis status.

- *Reference status*

For each reference the following extra information is available with full verbosity:

- If the reference does not exist (has been removed)
- If it is not unique
- If it is a named object
- If the reference has been analyzed more recently

### Revision Control

There is no integrated revision control system in the TTCN suite. Since normal visible files are used to store the TTCN documents it is easy to integrate the TTCN suite in a revision control system like SCCS or RCS. The `.itex` file format, being binary, is however not very suitable for that, it is better to use the TTCN-MP format to store in the revision control system.

Increased functionality is obtained if not only the TTCN-MP file is stored. The output from the Reporter contains useful information like modification dates and cross-reference lists.

By applying the following procedure whenever a check-point is desired the revision system becomes even more powerful:

1. Select the entire document.
2. Use the Convert to MP command to create a TTCN-MP file. Store it as e.g. `Test-Suite.mp`.
3. Use the Reporter, setting relevant options, to create a report. This report could now include modification date, cross-reference lists, etc. The *formatting* option *tab* should be chosen. This means that each TTCN object will occupy one line.
4. Store the log-window created by the reporter as e.g. `Test-Suite.rpt`.
5. Allow both files to be handled by the revision control system

It is now easy to obtain useful information about the difference between revisions by simply comparing the stored report files. A compare could be made using e.g. `scs diff` (if SCCS is used).

## Comparing TTCN Documents

The Compare tool is used for comparing two TTCN documents with respect to finding *similarities* that exist between them, i.e. two TTCN objects that have the same name. These objects need not necessarily be of the same type.

The Compare tool is accessed from a Browser and is applied to selections of items in that Browser.

Compare will check either only for similar names or for names **and** object type. The comparison can also include the contents of tables if required.

The Compare tool can also check the group structure of TTCN documents for similarities, i.e. it will indicate Test Group, Test Step Group and Default Group paths that are the same in both TTCN documents.

Any similarities that are found may be logged in a log file. There are three levels of log verbosity: *quiet*, *some* and *full*. Execution of the Compare tool may result in a new selection, which can be used to create a separate Browser if so wished.

One of the main uses of this tool will be to help remove conflicts from two TTCN documents prior to performing a merge operation.

### Note: UNIX only

The Compare tool is only available on UNIX.

## Comparing by Using the Compare Tool

To use the Compare tool, you need two TTCN documents to operate on: the *destination* document and the *compare* document.

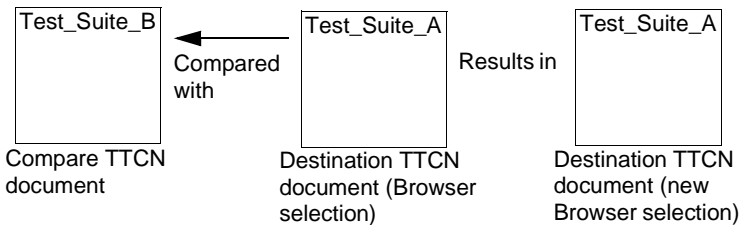


Figure 194: How Compare works

## Comparing TTCN Documents

---

A selection from the destination document may be compared with the compare document. If any similarities are found between the two TTCN documents this will result in the modification of the original selection in the destination TTCN document.

### **Tools > Compare**

Compares a Browser selection from one TTCN document with another TTCN document with respect to similarities. Make the selection in a destination TTCN document Browser and call the Compare tool from the same Browser.



*Figure 195: Compare Documents dialog*

**Log Device**

Controls the log device for the Compare tool. The default value is *Screen*, but the log can be directed to a named file by choosing *File* or turn it off altogether by choosing *None*.

For a full description of this command see “The TTCN Suite Logs” on page 6 in chapter 1, *User Interface and Basic Operations*.

**Documents**

Presents a list of currently open documents. For Compare to work correctly at least two documents must be present in this list. The document name in bold italic font is the document from which the Compare tool was called, i.e. the destination document. The user must choose another document in this list, i.e. the compare document, in order to do a comparison.

**Example 197**

---

In the dialog above the destination document is `Test_Suite_A` and hence the compare document may only be `Test_Suite_B`.

---

**Action**

Specifies the effect that the Compare operation will have on the original selection in the destination TTCN document.

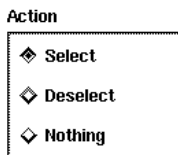


Figure 196: Compare: Action

- *Select*

All items in the original selection that are similar to items in the compare TTCN document will **remain selected**, otherwise the items will be deselected.

# Comparing TTCN Documents

---

In other words, the result of the Compare operation will result in a selection that shows the similarities between the destination and the compare TTCN documents.

- *Deselect*

All items in the original selection that are similar to items in the compare TTCN document will be **deselected**, otherwise the items will remain selected.

In other words, the result of the Compare operation will result in a selection that shows the differences between the TTCN documents.

- *Nothing*

Means that the Compare operation will have no effect on the destination selection. However, the results of the Compare operation will still be recorded in the log, if this option is set.

### **Verbosity**

Sets the verbosity level for the log file.

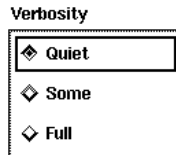


Figure 197: Compare: Verbosity

- *Quiet*

Turns the log off. No information will be recorded in the log, even if the *Log Device* is on.

### **Note:**

Using the *Nothing* action together with the *Quiet* option is rather meaningless!

- *Some*

Causes the names of all identically named objects to be printed in the log.

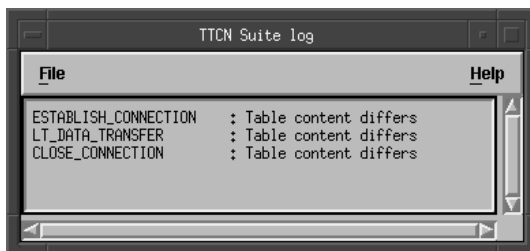


Figure 198: Compare: log (Verbosity: some)

If the *Content* switch is on, the log will also indicate that a difference has been detected.

- *Full*

Causes the names of all identically named objects to be printed in the log. However, if the objects are of different types this will be indicated as well.

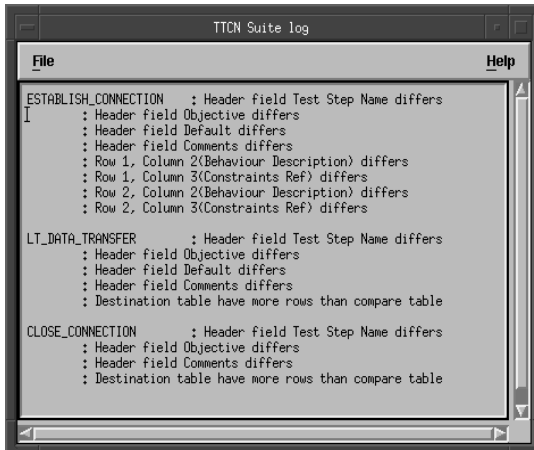


Figure 199: Compare: log (Verbosity: full)

If the *Compare Content* switch is on, the log will also indicate in detail the differences detected.



# Comparing TTCN Documents

---

## **Compare Content**

Extends the scope of the comparison to include the contents (down to the level of a single character) of the tables as well. Thus, two objects are considered to be identical if their names, object types and entire contents are identical. If the verbosity is full the differences are detailed.

## **Compare structure**

Extends the scope of the comparison to include the Test Group, Test Step Group and Default Group references in the relevant behaviour tables. This option is only available when the Compare content option is chosen.

## **Comparing by Using itexdiff**

A separate program, `itexdiff`, may also be used to quickly compare the contents of two TTCN suite files. The tool is less powerful than the Compare tool, but it may be very useful when there is a need for quickly comparing two revisions of the same TTCN document for equality.

The program is run like:

```
itexdiff FooSuite.itex /home/user1/FooSuite.itex
```

This will test the content of the two TTCN documents for equality and report the result in a printout. To get a more elaborate compare, the `-d` switch may be added like:

```
itexdiff -d FooSuite.itex /home/user1/FooSuite.itex
```

This will give the output of running the program `diff` on the MP representation of the content of the two documents.

Running the program without any arguments will print out a terse usage message.

## **Using Compare Before Merging Two TTCN Documents**

The Merge tool will only work if the two TTCN documents to be merged **do not conflict**. A conflict occurs if any TTCN objects in the TTCN documents have the same name.

One of the first things Merge does is to compare the two TTCN documents. If any conflicts are found it will be notified and the Merge oper-

ation will not be initiated. Use the Compare tool to find and resolve the conflicts.

For a full description of this command see [“Merging TTCN Documents”](#) on page 1137.

# Merging TTCN Documents

The Merge tool is used for merging one TTCN document (complete or partial) into another TTCN document.

A useful application of the Merge tool is to use the Selector tool and the Convert to MP tool to extract a complete test case (the Test Case Dynamic Behaviour together with its declarations, constraints etc.) and merge it with another TTCN document.

### Note: UNIX only

The Merge tool is only available on UNIX.

## Preparing for a Merge

The merge tool requires that two TTCN documents are opened. We shall call them the *merge* TTCN document and the *destination* TTCN document or selection.

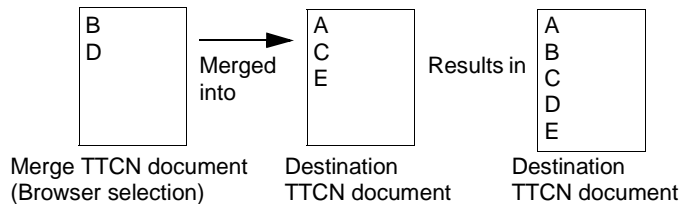


Figure 200: Shows how Merge works

Merge will only work if the two TTCN documents **do not conflict**. A conflict occurs if any selected TTCN object in the merged TTCN document has the same name as any TTCN object in the destination document.

One of the first things Merge does is to compare the two documents. Any conflicts that are found will be notified and the Merge operation will be aborted. Use the Compare tool to find and resolve the conflicts.

For a full description of this command see [“Comparing TTCN Documents”](#) on page 1130.

## Merging Two TTCN Documents

Once the selected objects in the merge TTCN document do not conflict with the destination TTCN document, the Merge tool can be used.

### *Tools > Merge*

Merges a selection into another TTCN document. Make the selection in the merge TTCN document and call the merge tool to merge the selected nodes into the destination document.

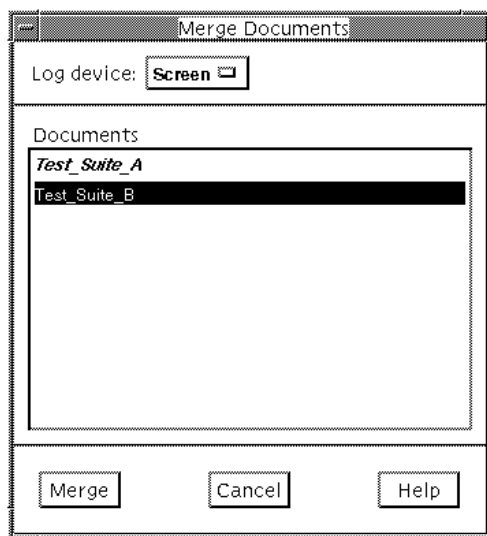


Figure 201: Merge Documents dialog

- *Log device*

Enables the user to define the log device for the Merge tool. The default value is *Screen*, but the user can direct the log to a named file by choosing *File* or turn it off altogether by choosing *None*.

For a full description of this command see [“The TTCN Suite Logs” on page 6 in chapter 1, \*User Interface and Basic Operations\*](#).

- *Documents*

Presents a list of currently open documents. For Merge to work correctly at least two documents must be present in this list. The document name in bold italic font is the document from which the Merge tool was called, i.e. the merge document. The user must choose another document in this list, i.e. the destination document, in order to do a merge.

In the dialog above the destination document is `Test_Suite_B` and the merge document is `Test_Suite_A`.

### Creating Documents by Using the Merge Tool

The Merge Tool is not only useful when merging two documents together. It can also be used in the creation of a new document that needs to obtain some information from another document. The following procedure can be useful:

1. Create a new (empty) document.
2. Open the document containing objects that are to be copied.
3. Select the desired objects in the Browser (possibly aided by the Selector tool).
4. Merge them into the empty document.

There are also alternate ways of doing this. Copy/Paste between the documents or using the Convert to MP tool and the Convert to GR tool or the Merge from MP tool.

## Merging from MP Files

Use the menu choice *Merge from MP* or *Autolink Merge* for merging the content of an MP file into a TTCN document. For additional information on merging an Autolink generated MP file, see [“Merging TTCN Test Suites in the TTCN Suite”](#) on page 1360 in chapter 36, *TTCN Test Suite Generation*.

Four steps are involved in a merge from MP:

1. It first converts the MP file to a temporary GR format TTCN document, see [“Importing a TTCN-MP Document”](#) on page 1149 for details.
2. The second step is to use the Merge tool to copy all the content of the new document into the existing document, see [“Merging TTCN Documents”](#) on page 1137 for details (note the conflict resolution difference).
3. The third step is to remove the temporary document.
4. The optional fourth step is to analyze the destination document (this last step is mandatory if called under the name *AutoLink Merge*)

The Merge from MP tool requires that one TTCN document is open. We shall call it the *destination* TTCN document. This is where the merge from MP is called from.

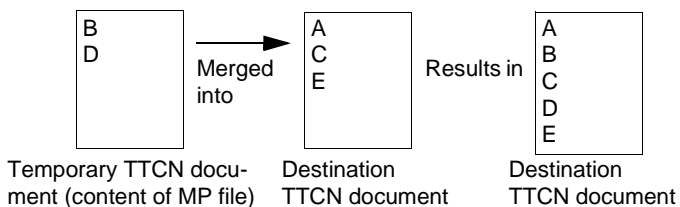


Figure 202: How a merge from MP works

Merge from MP will only work as depicted above if the two TTCN documents **do not conflict**. A conflict occurs if any TTCN object in the MP file has the same name as any TTCN object in the destination document. If such a conflict is detected, the conflicting object in the MP file will be skipped and the merge process continued.

## Merging from MP Files

---

Merging of constraint tables are handled specially. For example, an MP file may contain a TTCN ASP Constraint `constraint1` that refers to the type `type1`, where the `type1` Type table is of the incompatible type TTCN PDU TypeDef. This will make the merge process insert a copy of `constraint1` as a table of type TTCN PDU Constraint instead of the original type<sup>1</sup>. There are, however, limitations to this “type conversion”. The conversion will not convert an ASN.1 Constraint to a TTCN Constraint nor vice versa.

### ***Tools > Merge from MP***

#### ***SDT Link > Autolink Merge***

The menu choice *Merge from MP* in the *Tools* menu and *Autolink Merge* in the *SDT Link* menu makes it possible to merge the content of an MP file into the current TTCN document.

---

1. This feature does not exist in the regular Merge tool.

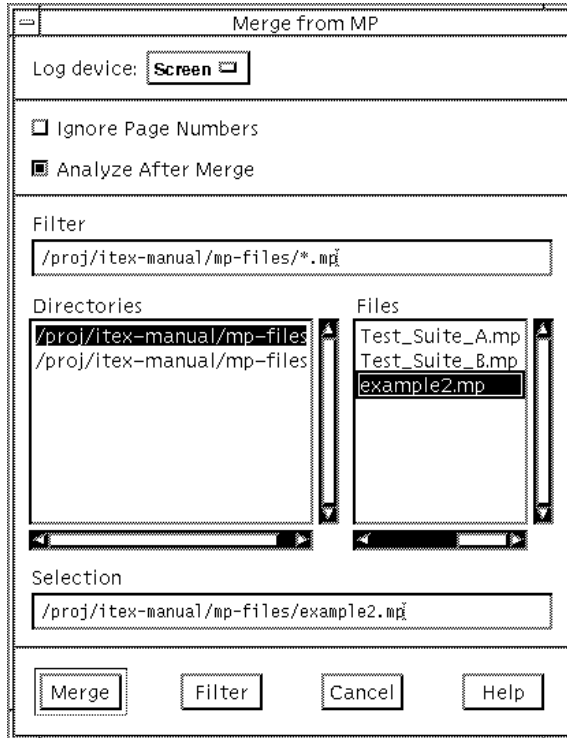


Figure 203: The Merge from MP dialog

- *Log device*

Enables you to define the log device for the merge from MP. The default value is *Screen*, but the user can direct the log to a named file by choosing *File* or turn it off altogether by choosing *None*.

For a full description of this command see [“The TTCN Suite Logs” on page 6 in chapter 1, \*User Interface and Basic Operations\*](#).

- *Ignore Page Numbers*

Ignore any page number information in the MP file if set.



## Merging from MP Files

---

- *Analyze After Merge*  
Runs the Analyzer on the whole destination document after successful merge if set. This option is not possible to disable when using *AutoLink Merge* and so is not displayed in that dialog.
- *Filter*  
Sets the filter for the files that will be displayed in the *Files* list. By convention TTCN-MP files are given the suffix `.mp`.

### Example 198

---

The name filter `*.mp` will cause only those files whose names end with `.mp` to be displayed, e.g. `example.mp`

---

- *Directories*  
Lists the directories in the current file system directory. To change directory, point to the required directory name and double-click.
- *Files*  
Lists the files (if any) that are in the current directory and which match the filter.
- *Selection*  
Displays the selected document.
- *Merge*  
Terminates the dialog and executes the merge from MP.
- *Filter*  
Updates the *Files* list according to the current filter.

## Exporting a TTCN Document to TTCN-MP

TTCN-MP – the textual notation (*machine processable*) – can be used when you want to import a TTCN document into a non-TTCN suite tool or make backups.

By using the menu command *Convert to MP*, you can convert a TTCN-GR document to TTCN-MP. However, note that it is also possible to simply use *Save Document As* for changing the format. The difference between the two menu commands is that *Convert to MP* gives the option to save as TTCN Suite MP or standard MP, while *Save As* produces TTCN Suite MP. How this differs from the standard MP format is described in [“The TTCN Suite MP format” on page 1146](#).

There are no restrictions on the analysis state of the TTCN document with regards to its correctness, i.e. it will generate TTCN-MP even for TTCN documents that are neither complete nor error-free. This implies that the generated MP file may be, but is not necessarily, conformant to the standardized TTCN-MP format.

However, to ensure that a TTCN document that is output in TTCN-MP by the TTCN suite can be read by a non-TTCN suite tool, it is necessary to make sure that the TTCN document is correctly analyzed, that the test suite overview have been recently updated. You also have to select everything in the document before convert to MP. The TTCN suite format option in the dialog will in most such cases have to be non-selected.

Convert to MP, invoked from the Browser, works on Browser selections, i.e. it is possible to output selected items (e.g. a single test case or just the constraints). Convert to MP will add the necessary keywords to the MP file to make it into a TTCN-MP that can be converted back into GR format.

An optional but non-standard TTCN-MP for compact tables and ASN.1 references is supported. Additional fields in dynamic tables (fields which are transferred to test suite overview tables) are also supported. This format is the recommended format for transferring TTCN document between the TTCN suite instances, and also for making backups of the TTCN documents.

Convert to MP is accessible from the Browser *File* menu and from the Organizer *Generate* menu.

### Converting to TTCN-MP in the TTCN Suite

The items selected in the Browser will be converted to TTCN-MP. Selections can be arbitrary – the TTCN suite will add the necessary keywords to the MP file to make it into a TTCN-MP that can be converted back into GR format.

To convert an entire TTCN document, do a *Select All* and then select *Convert to MP* from the *File* menu, or invoke this operation from the Organizer. For more information, see “[Convert to MP \(TTCN\)](#)” on page 133 in chapter 2, *The Organizer*.

For example, to convert a complete test case, use the Selector, together with the *References recursively* option and then apply *Convert to MP* on the resulting selection.

#### ***File > Convert to MP***

Presents a dialog in which you can convert the selected items to MP.

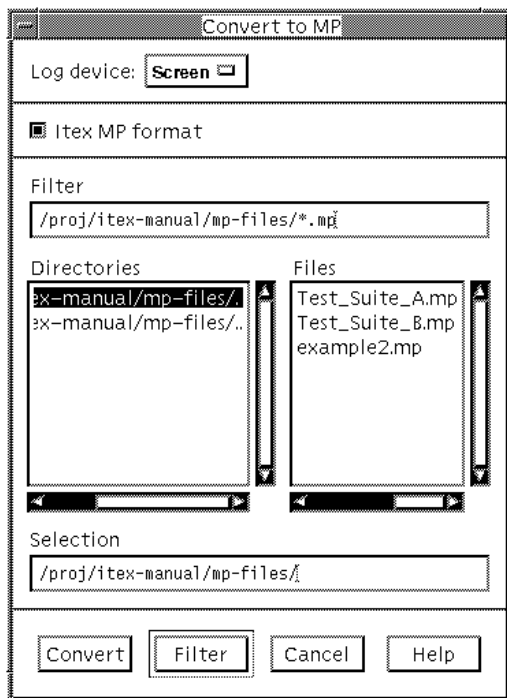


Figure 204: Convert to MP file dialog

### Log Device

Controls the log device for the Convert to MP tool. The default value is *Screen*, but the log can be directed to a named file by choosing *File* or disabled by choosing *None*.

For a full description of this command see [“The TTCN Suite Logs” on page 6 in chapter 1, \*User Interface and Basic Operations\*](#).

### The TTCN Suite MP format

Enables the user to convert a TTCN document to a non-standard MP with the following extra information:

- *Compact tables*

Is intended for TTCN documents where the Test Case Dynamic Behaviours (i.e. compact test cases) table has been used. Check this option to generate an MP file that contains additional keywords that specifically identify compact tables. If this option is not used the TTCN suite will output a standard TTCN-MP file where each test case in the compact test cases table is represented as a single test case (i.e. the information about the compact table is lost).

- *Additional fields in dynamic tables*

If the *ITEX MP format* option is used, the TTCN suite will convert the additional Selection Ref and Description fields in the behaviour tables.

- *ASN.1 reference tables*

The additional field in ASN.1 type by reference and Test Suite Constant Declarations By Reference tables will be converted if this option is checked. These additional fields contain the definition of the ASN.1 type and value.

- *Import tables*

The additional column fields in Import tables will also be converted if this option is checked.

### ***Filter***

Sets the filter for the files that will be displayed in the *Files* list. By convention TTCN-MP files are given the suffix `.mp`.

### **Example 199**

---

The name filter `*.mp` will cause only those files whose names end with `.mp` to be displayed, e.g. `example.mp`

---

### ***Directories***

Lists the directories in the current file system directory. To change directory, point to the required directory name and double-click.

**Files**

Lists the TTCN suite TTCN-MP files (if any) that are in the current directory and which match the filter.

**Selection**

Displays the selected document.

**Filter**

Updates the *Files* list according to the filter.

**Exporting by Using itex2mp**

A separate program, `itex2mp`, may be used to quickly export the content of a TTCN suite file. The generated MP format includes the TTCN suite-specific extensions.

The program is run like:

```
itex2mp FooSuite.itex FooSuite.mp
```

This will export the content of the TTCN document into the specified MP file. You may notice that the program has the “style” of a specialized `cp` program.

Running the program without any arguments will print out a terse usage message.

# Importing a TTCN-MP Document

By using the menu choice *Convert to GR* in the Organizer, you can convert a TTCN-MP document to TTCN-GR and open it in the TTCN suite. However, note that it is also possible to simply open an MP document.

The Convert to GR tool is flexible, it will read any TTCN-MP file that has been successfully generated by the Convert to MP tool, even if the MP file only contain parts of a test suite. Convert to GR will also read TTCN-MP from other products which support TTCN assuming that this TTCN-MP follows the ISO standard.

For a full EBNF supported by the Convert to GR tool, see [“The TTCN-MP Syntax Productions in BNF” on page 1530 in chapter 39, \*Languages Supported in the TTCN Suite\*](#).

An optional but non-standard TTCN-MP for converting TTCN documents that use compact tables and ASN.1 references with an explicit definition, is also supported.

## Converting a TTCN Document

For a full description of this command see [“Convert to GR \(TTCN\)” on page 135 in chapter 2, \*The Organizer\*](#).

## MP File Format Problem

When converting certain TTCN documents, a problem with transferring the information (e.g. Description) in the Overview tables to the tables in Dynamic Part, may occur.

The TTCN standard allow path specifications to optionally include the document name first. This has the unfortunate effect that if the TTCN document contains a top level group with the same name as the TTCN document there is, in general, no way of knowing if the first part of the paths is a group identifier or the TTCN document identifier.

The TTCN suite assumes that if the first part of the path is equal to the document name, it is the optional document name and, when converting, strips it away. When exporting, the document name is always added to the front of all paths. That way the TTCN suite is always able to convert the MP files it exports. Note that ITEX 2.0 did not add the docu-

ment name at export, and therefore the problem described here may also apply when converting MP files exported by ITEX 2.0.

If the TTCN document contains top level group identifiers equal to the document name, and the TTCN suite is unable to resolve the paths, temporarily change the document name in the MP file and change it back once inside the TTCN suite.

## Converting Fields Containing the Dollar Character

Normally the Convert to GR tool ignores the content of the fields of tables in the MP file but, to allow converting of non-BoundedFreeText fields (e.g. TS\_VarValue field) with embedded dollar characters (e.g. “\$”) the Convert to GR tool has been enhanced with a simple syntax checker of the fields. This has the implication that it is no longer possible to convert TTCN documents containing unmatched single or double quotes in non-BoundedFreeText fields, i.e. Convert to GR is less tolerant about syntax errors in the fields.



# Generating the Test Suite Overview Tables

The TTCN suite generates the contents of the Test Suite Overview tables (i.e. The Test Suite Structure, Test Case Index, Test Step Index and Default Index tables) and the TTCN Module Overview (i.e. The TTCN Module Structure and the index tables).

These tables are **not** editable. The extra information that they require (such as references to selection expressions and descriptions) are entered at the corresponding table rather than in the Test Suite Overview tables. This ensures consistency between the TTCN document and the contents of the Test Suite Overview.

## Selection Expressions

The Test Case Dynamic Behaviour table displayed by the TTCN Table Editor has an additional field in the table header called: *Selection Ref*.

This field is used to hold the reference to a Selection Expression (if any) declared in the Test Case Selection Expression Definitions table. This reference will then appear in the *Selection Ref* column of the Test Case Index when the index is generated using the Generate tool.

If the Selection Reference needs to be edited this must be done in the relevant Test Case table and not in the Test Case Index table (which cannot be edited). The Test Suite Overview (or the TTCN Module Overview) must be re-generated in order to update any changes.

The additional *Selection Ref* field in the Test Case header is **not** printed in the TTCN-GR form. It is however exported in the TTCN suite specific MP format.

## Test Case, Test Step and Default Descriptions

The Test Case, the Test Step and the Default Dynamic Behaviour tables displayed by the TTCN Table Editor have an additional field in the table header called *Description*.

The *Description* field is used to hold the description of the behaviour table, which in many cases may simply be a copy or a shortened form of the Test Purpose or Test Step or Default Objective. This description will then appear in the Description column of the relevant index when it is generated using the Generate Overview tool.

If the Description needs to be edited this must be done in the relevant behaviour table and not in the index tables (which cannot be edited). The Test Suite Overview (or the TTCN Module Overview) must be re-generated in order to update any changes.

The additional Description field in the Test Case, Test Step and Default headers is **not** printed in the TTCN-GR form. It is however exported in the TTCN suite specific MP format.

## The Group Table

The TTCN Browser supports additional non-standard tables for groups. These tables are: the *Test Group* table, the *Test Step Group* table and the *Default Group* table. **None** of these tables are printed in the TTCN-GR form.

The Test Group table has two fields in its header that are used to hold information for the Test Suite Structure table. These fields are: *Selection Ref* and *Test Group Objective*.

The *Selection Ref* field is used to hold a reference to a Selection Expression (if any) declared in the Test Case Selection Expression Definitions table. This reference will then appear in the Selection Ref column of the Test Suite Structure when it is generated using the Generate tool.

The *Test Group Objective* field is used to hold the description of the Test Group. This description will then appear in the *Test Group Objective* column of the Test Suite Structure when it is generated using the Generate tool.

If either the Selection Reference or the Test Group Objective needs to be edited this must be done in the relevant Test Group table and not in the Test Suite Structure table (which cannot be edited). The Test Suite Overview must be re-generated in order to update any changes.

All of these fields are exported in the TTCN suite specific MP format.

## Generation of the Test Suite Overview and Indices

The Generate tool is available from the *Tools* menu in the Browser and the *Tools* menu in the Table Editor. Unlike other tools it does **not** require a selection: it always applies to the entire TTCN document and its action is the same whether it is invoked from a Browser or from a Table Editor.

### *Tools > Generate SO*



Generates the Test Suite Overview (or the TTCN Module Overview) tables (i.e. The Test Suite Structure, TTCN Module Structure, Test Case Index, Test Step Index and Default Index tables). It makes use of the additional Group tables and the additional Selection Ref and Description fields in the behaviour tables.

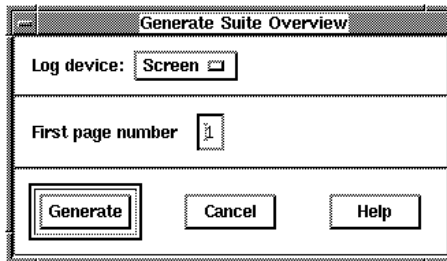


Figure 205: Generate Suite Overview dialog

- The *Log Device* option controls the log device for the Generate tool. The default value is *Screen*, but the log can be directed to a named file by choosing *File* or turn it off altogether by choosing *None*.

For a full description of this command see “[The TTCN Suite Logs](#)” on page 6 in chapter 1, *User Interface and Basic Operations*.

- The *First Page Number* option enables setting the first page number of the TTCN document. The default value is 1.

## Crash Recovery

In the event of a TTCN suite crash, it is possible to recover up-to-the-minute changes. The auto-save content is stored in two “comma files”: `filename,s` and `filename,t`, where `filename` is derived from the first five characters of the name of the file they are associated with. The files can be found in the designated temporary directory, usually `/var/tmp`. To be able to open those files, you need to concatenate them into a `.itex` format file. This is accomplished with the `iconcat` program.

The program is run like:

```
iconcat /var/tmp/FooSuJAAa002189 FooSuite-saved.itex
```

This will concatenate the content of the TTCN suite files

`/var/tmp/FooSuJAAa002189,s` and

`/var/tmp/FooSuJAAa002189,t` into the `FooSuite-saved.itex`

ITEX file. Note that the program ignores any existing

`/var/tmp/FooSuJAAa002189` file, despite the argument syntax.

You may notice that the program has the “style” of a specialized `cp` program. Running the program without any arguments will print out a terse usage message.

# Key and Button Bindings

Key	Action
Ctrl+Shift+space	Toggles the select status of all nodes in the subtree.
Ctrl+space	Toggles the select status of the focused node.
Shift+space	Set the select status of all nodes in the subtree. Remove select status on all others.
Space	Set the select status of the focused node. Remove select status on all others.
t	Open an editor on the focused node.
osfHelp	Get help on Browser.
c	Hide subtree.
e	Make children of the focused node visible.
Shift+e	Make whole subtree visible.
r	Rename the focused node.
Ctrl+c	Copy the focused node (for later paste).
Ctrl+x	Cut the focused node (for later paste).
Ctrl+v	Paste the content of the paste buffer before/above the focused node.
Ctrl+Shift+osfInsert	Add a new child to the focused node. It is placed last in the child list.
Ctrl+osfInsert	Add a new sibling before/above the focused node.
Ctrl+Meta+osfInsert	Add a new sibling after/below the focused node.
Ctrl+Shift+G	Add a new group child to the focused node. It is placed last in the child list.
Ctrl+g	Add a new group sibling before/above the focused node.
Ctrl+Meta+g	Add a new group sibling after/below the focused node.

Key	Action
F2	Move down in tree such that if we start on root node we come to “Test Cases”.
Ctrl+/ Ctrl+\	Select all nodes in browser. Deselect all nodes in browser.
osfUp	Move focus to previous visible node.
osfDown	Move focus to next visible node.
osfLeft	Move focus to parent of the focused node.
osfRight	If the children of the focused node are hidden, make them visible. Then move focus to first child.
Shift+osfUp	Move focus to previous node at same level.
Shift+osfDown	Move focus to next node at same level.
Shift+osfLeft	Move focus to parent of the focused node, and hide the subtree.
Shift+osfRight	If the children of the focused node are hidden, make them visible. Then move focus to last child.
osfBeginLine	Move to first visible node of browser.
osfEndLine	Move to last visible node of browser.
Shift+F10	Show node popup menu.
Ctrl+Z	Run “Analyze & Stop” on page 1183 in chapter <a href="#">27, Analyzing TTCN Documents (on UNIX)</a>

## Key and Button Bindings

---

<b>Mouse button (OSF Names)</b>	<b>Action</b>
Button1	Set focus to the node and set selection status of the node. Remove select status on all others.
Ctrl+Button1	Set focus to the node and toggle selection status of the node.
Shift+Button1	Set focus to the node and set selection status of the subtree. Remove select status on all others.
Ctrl+Shift+Button1	Set focus to the node and toggle selection status of the subtree.
Button1 twice	Set focus to the node and start an editor on that node if that is at all reasonable. If the node is a static node, make its children visible or hidden. Remove select status on all nodes.
Ctrl+Button1 twice	Set focus to the node and start an editor on that node.
Button2	Move focus to the node.
Button3	Show popup menu.

