Chapter

2 *The Organizer*

The Organizer is the main tool in Telelogic Tau. When you invoke a Telelogic Tau tool, the Organizer is started and displayed on the screen.

This chapter contains a reference manual for the Organizer; the functionality it provides, its menus, windows and symbols.

Overview

Terminology

The following basic terminology is used throughout this chapter:

- *Document*

  A *document* is a file containing information that can be edited in an editor available through the use Telelogic Tau. Examples of documents in the Organizer are SDL diagrams, MSC diagrams, TTCN documents, and text documents.

- *Generic Document*

  Files containing document types that the Organizer does not normally handle can be included in the Organizer by using a *generic document* symbol and connecting that symbol to the file. The Organizer uses the file extension and the preference variable Organizer*GenericCommand to determine a command to perform when the user tries to edit a generic document.

- *Diagram*

  A *diagram* is a graphical document that only can be edited in a graphical editor. Diagrams are either SDL diagrams, MSC diagrams, HMSC diagrams, Object Model (OM) diagrams, or State Chart (SC) diagrams.

- *TTCN Document*

  A *TTCN Document* signifies a document of type Test Suite, Modular Test Suite, TTCN Module, or TTCN Package.

- *SDL System*

  An *SDL system* in Telelogic Tau signifies the topmost SDL diagram with all its sub diagrams, and all used SDL packages with their sub diagrams. In the normal case, the topmost diagram is an SDL system diagram, but the Organizer allows all types of SDL diagrams to be the topmost diagram, as long as the standard hierarchical rules of SDL diagrams are respected.

- *TTCN System*

  A *TTCN system* in Telelogic Tau signifies the topmost TTCN document with all its sub TTCN documents.

- *System*

  Not to confuse with an SDL or TTCN system, a *system* in Telelogic Tau is a set of documents that, according to your view, are related and thus managed by the Organizer. You decide which documents make up a system and how they are related and grouped together in the Organizer.

  A system can contain documents for analysis, design and testing of one or more SDL and/or TTCN systems. The documents can be textual requirements, analysis and design diagrams, test specifications, source code in different forms, and other types of related documentation.

  In the normal case, one system in the SDL suite contains one SDL system. But for special purposes it is possible to have more than one SDL system in the system context. Such a purpose could be when working with communicating simulators on UNIX where the source diagrams of each system are required to show and trace the graphical source symbols.

- *Root Document*

  The topmost document in a document structure, or a stand-alone document, or a top-level document in a module. For an SDL system, the topmost diagram in the system, usually an SDL system diagram.

  SDL package diagrams and macro diagrams are special in the sense that they are used by an SDL system diagram. Therefore, they are also placed at the root level and are considered root diagrams.

- *Association*

  An *association* is a link between two documents. Any document can be associated with any other document; a typical example is to associate an MSC diagram with a related SDL diagram. In the document structure, an association symbol is included to the associated document. This symbol is handled like a document, even though it simply represents a document residing somewhere else.

- *Chapter*

  The drawing area of the Organizer is divided into several chapters. You may freely add, remove, and rename chapters, as well as rearrange the order of the chapters. Each chapter has an associated chapter level. It is possible to set the start chapter number.

- *Module*

  A concept specific to the Organizer, used for freely grouping root documents together into a document structure. The user can use modules as a kind of scope unit. Modules have no corresponding concept in SDL, MSC or TTCN terminology. Note, however, that a TTCN Module is not the same concept as a module in the Organizer.

- *Logical Diagram Name*

  The Organizer can identify diagrams in the system by their logical diagram names. The logical diagram name may contain the entire qualified name of the diagram in a specific format. The connection between logical diagram names and physical file names is explicit and under user control.

- *System File*

  A file containing information about the structure and state information of a system as seen by one Telelogic Tau user. Organizer user settings and viewing options are also stored in this file. The system file is described further in "System File" on page 184.

- *Control Unit File*

  A file that contains a *control unit*, i.e. structure information for a part of a system, common to all users that work with the system. The purpose is to provide workgroup support and a means to put a system under revision control. The user has full control of what parts of a system should be managed as control units. The control unit file is described further in "Control Unit File" on page 195.

- *Link File*

  A file that contains the endpoint and link information, which is managed by the Link Manager. The link file is described further in "The Link File" on page 485.

- *Source Directory*
- *Target Directory*

  The source directory in Telelogic Tau specifies where new documents that you have created, are saved by default, and where to read from when opening and converting documents. **On UNIX,** the source directory is also the directory normally shown when you click *Current* in a standard file selection dialog or a standard directory selection dialog. For more information, see "File Selection Dialog" on page 30 and "Directory Selection Dialog" on page 31 in chapter 1, *User Interface and Basic Operations*.

  The target directory specifies where to put generated files.

  These directories are set in the Organizer, see "Set Directories" on page 70. They are also saved in the system file.

- *Footer File*
- *Header File*

  Footer and header files are used to define how footers and headers should look like when pages are printed. The footer and header file symbols makes it easy to edit the appearance of the footer or header using a text editor. The footer and header file format is described in "Footer and Header Files" on page 335 in chapter 5, *Printing Documents and Diagrams*.

Organizer User Interface

The Organizer consists of two windows: a main window and a log window. For a general description of the user interface, see chapter 1, *User Interface and Basic Operations*.

The main window of the Organizer is also the main window of the Telelogic Tau environment. It manages the current system structure and is responsible for invocation and termination of other tools in Telelogic Tau.

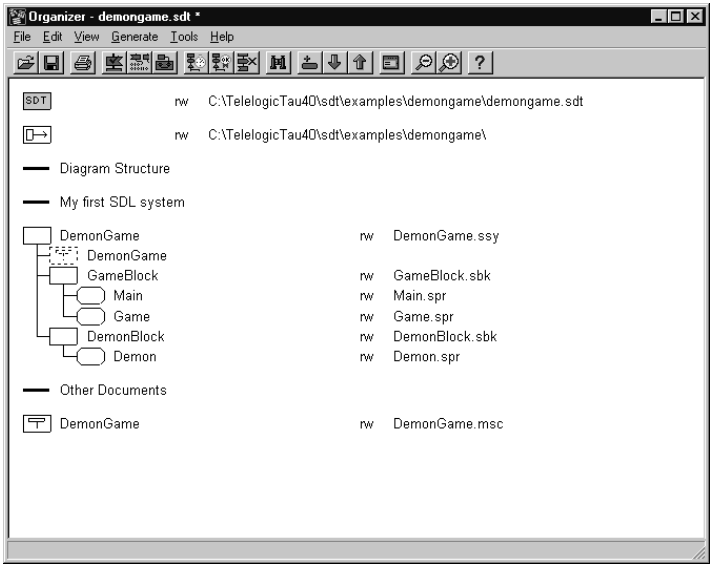

Figure 9: The Organizer main window

The main window exists in one instance only. If more than one Organizer is required, another Telelogic Tau tool must be started. For special purposes, the Organizer can handle more than one TTCN or SDL system simultaneously.

The log window is a text window used for displaying output during analysis and code generation and for displaying general logs. The window is described in "Organizer Log Window" on page 179.

Drawing Area

In the Organizer's drawing area, the files in the system and their structure are presented graphically using icons.

Presentation Modes

The Organizer can present the system information in two different ways graphically:

- As an *indented list*, i.e. a compact, line-oriented list of icons using indentation to suggest the structure (see Figure 10).
- As a *vertical tree*, which is depth- or level-oriented and requires a larger amount of space (see Figure 11).

The user can change the amount of information and the way it is presented in the drawing area with the menu choice *View Options* in the *View* menu. See "View Options" on page 105 for more information.

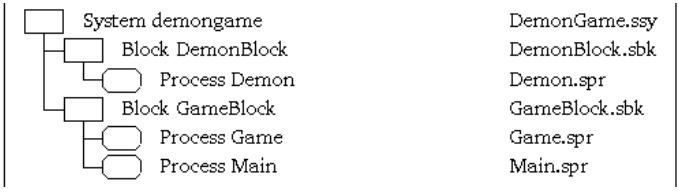

Figure 10: Indented list mode

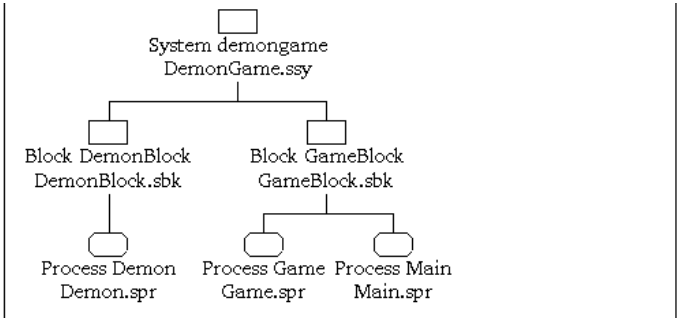

Figure 11: Vertical tree mode

Chapters

The *chapter* concept can be used to define a collection of documents. There are no restrictions on what kind of documents that may be placed in a certain chapter, or on the order of documents in a chapter; this is for the user to decide. Chapters can further be used to structure a print-out of the system included in the Organizer view. More information about this can be found in "Advanced Print Facilities" on page 348 in chapter 5, *Printing Documents and Diagrams*.

A chapter has a name, and possibly a number. A chapter is shown as a thick horizontal line in the drawing area, together with its number and name. A chapter symbol can be placed inside a Module, as well as inside an SDL diagram structure and a TTCN Module structure.

In addition to the line icon, it is possible to have the Organizer show the text "Chapter" ahead of the number and name; see "Icon Names and Types" on page 55.

Chapter Levels

A chapter has an associated chapter level, ranging from 0 to 4, which is reflected in the numbering of the chapter:

- Level 0: a chapter without a number (only a name)
- Level 1: a chapter with a single number: 1, 2, 3, etc.
- Level 2: a chapter with two numbers: 1.1, 1.2, 1.3, etc.
- Level 3: a chapter with three numbers: 1.1.1, 1.1.2, 1.1.3, etc.
- Level 4: a chapter with four numbers: 1.1.1.1, 1.1.1.2, 1.1.1.3, etc.

Numbered chapters are auto-numbered by the Organizer. The initial number for the first chapter in the drawing area can be specified. The initial level of a chapter is specified when the chapter is created, but can later be changed.

Chapter Introduction Text

A chapter can be associated with a text, which can be regarded as an introduction text to the chapter. When the chapter is included in a print-out, this text will be inserted on a separate page corresponding to the position of the chapter symbol.

The text is saved in a text file, which the chapter symbol is connected to.

Default Chapters

There is a default set of unnumbered chapters, known as the *basic Organizer view*, intended for different kinds of documents. This view appears when creating a new system file and contains the following chapters:

- *Chapter Analysis Model*
- *Chapter Used Files*
- *Chapter SDL System Structure*
- *Chapter TTCN Test Specification*
- *Chapter Other Documents*.

The intended use of these default chapters is described below. The default set of chapters can be changed by editing the preference Organizer*Areas.

Chapter Analysis Model

The *Analysis Model* chapter is intended for documents used in a system's analysis, such as Object Model and State Chart diagrams, text requirement documents, MSC, HMSC and SDL requirement diagrams, etc.

Chapter Used Files

The *Used Files* chapter is intended for documents that are not generated by Telelogic Tau and that are used, included, or imported by other documents in other chapters. Such documents may be C/C++ header files, etc.

Chapter SDL System Structure

The *SDL System Structure* chapter is intended for the SDL diagram structure that normally is the primary information managed by the Organizer. The diagram structure contains a strictly hierarchical view of one or more root diagrams and their substructures. Most often only one diagram structure is used (a system) with one or more package or macro structures present as separate roots in the system. A characteristic of these diagrams is that they are directly involved in the analysis/code generation process. Multiple system roots are allowed, mainly to support the possibility of running communicating simulators with graphical trace.

Chapter TTCN Test Specification

The *TTCN Test Specification* chapter is intended for TTCN documents used for test specifications.

Chapter Other Documents

The *Other Documents* chapter is intended for documents that do not fit into any of the other chapters. It may contain SDL, MSC, and HMSC diagrams, text documents, etc. that are not directly part of an SDL system and that are not related to the other document structures in the Organizer.

Structure Icons

A few special icons in the Organizer are used for showing how document icons are structured, and for specifying where they are stored in the file system by default.

In addition to the icon, it is possible to have the Organizer show the type of the structure icon textually; see "Icon Names and Types" on page 55.

Different operations are applicable to different types of icons. Double-clicking an icon is described in "Double Clicks" on page 56, and associated popup menus are described in "Pop-Up Menus" on page 166.

System File
The name of the system file. This system file could be managed as a control unit (CM group) to allow revision control of the system file in a multiuser environment.

Link File
The name of the link file.

Control Unit File
This icon designates a control unit whose contents have not yet been resolved.

Source Directory
The name of the source directory.

Target Directory
The name of the target directory.

Header File
The name of the header definition file. Note that it is possible to connect a text file to this symbol, i.e. the symbol can act as a plain text symbol.

Footer File
The name of the footer definition file. Note that it is possible to connect a text file to this symbol, i.e. the symbol can act as a plain text symbol.

Chapter
A chapter in the drawing area. A chapter can have a number and a name and may be connected to a text file.

Module
Another way to define a collection of documents, within a chapter. A module has a name but no associated file name.

More
Indicates that a number of documents are hidden in the document structure (see "Hide" on page 105). The text associated with the icon indicates the number of hidden (additional) documents, e.g. "5 More".

Bookmark
Quick access to a place in an SDL Suite diagram or document via an SDT reference or to a place on the Web via a URL.

Document Icons

To visualize the different types of documents that are present in the user's system, a large number of graphical icons are used. A unique icon is provided for every document type that is commonly used in a system. In particular, each SDL diagram type (except packages, systems, macro definitions and operators) already have a given icon (the reference symbol specified in the Z.100 recommendation).

In addition to the icon, it is possible to have the Organizer show the type of the document textually; see "Icon Names and Types" on page 55.

Different operations are applicable to different types of icons. Double clicking an icon is described in "Double Clicks" on page 56, and associated popup menus are described in "Pop-Up Menus" on page 166.

Icon Types

The list below shows the icons that may be part of a system display in the Organizer:

Object Model Diagram icons


Object Model


State Chart

SDL Diagram icons


Package


System


Block


Substructure


Process


Service


Procedure


Macro


System type


Block type


Process type


Service type


Operator


SDL Overview

MSC Diagram icons


MSC


HMSC

Text Document icons

ASN.1 Text

Build Script (see "Build Scripts" on page 2572 in chapter 57, *The Cadvanced/Cbasic SDL to C Compiler*)

C Header

C Import Specification

C++ Import Specification

Plain Text. Note that connecting a plain text symbol to an external synonym file (*.syn) has a special meaning. See "Update Visibility" on page 101, "Include Expression" on page 1948 in chapter 44, *Using the SDL Editor* and "Supplying Values of External Synonyms" on page 2172 in chapter 51, *Simulating a System*.

Test Script. Used by Organizer>Tools>Simulator Test to execute test cases (in the form of input history scripts *.cui) in the simulator.

SDL PR

Word Text

TTCN Document icons

Modular Test Suite

Module

Package

Test Suite

Flat View

SDL Instance icons

System instance

Block instance

Process instance

Service instance

SDL Dashed icons

Block

Process

Service

Other icons

The icon for an association or a dependency is a dashed version of the icon of the referenced document. The example to the left shows an association referencing an MSC icon.

Generic Document

Index icon, representing a generated cross reference file that can be viewed in the Index Viewer.

Page (used for SDL, HMSC, State Chart, and OM diagrams)

Diagram and Document Icon States

The diagram and document icons can have different background colors indicating the state of the information entity associated with the icon. If the icon state is anything else than normal, additional information about the corresponding file is provided in the Status Bar when the icon is selected.

Normal
The normal state of the icon. Information is not modified.
All diagram and document icon types can be normal. This is also the background color used by all non-diagram/document icons.

Invalid
The connected file does not exist or is not a diagram file of the correct type. This state is only possible when connected to a file.
All diagram and document icons can be invalid.
For more information on invalid icons, see "Open" on page 59 and "The Drives Section" on page 187.

Mismatch
The SDL diagram is not referenced in the diagram file where this diagram is referenced in the Organizer, i.e. there is a SDL diagram structure mismatch between the system file and the diagram files.
This state is only possible when connected to a file. Mismatched diagrams that are unconnected are removed from the diagram structure in the Organizer.
SDL diagram icons that are not root diagrams can be mismatched.
The reason for a mismatch is that the reference symbol in the parent diagram states a diagram name different than the kernel heading in the mismatched SDL diagram.
For more information on mismatches, see "Open" on page 59 and "No Save" on page 63.

Dirty
The associated information entity is modified, but not yet saved. This state is only possible when connected to a file.
All diagram and document icons can be dirty.

Ordering

The order of root documents in a chapter is not fixed and can be changed by the user; see "Move Down" on page 178 and "Move Up" on page 178. Moving documents in the Organizer is also used for two other purposes:

- Grouping root documents in a module. Trying to move a root document "over" a non-collapsed module, will result in that the root document becomes a child symbol to the module. A document is also moved out of a module by moving the document up (or down) one or several times.
- Grouping TTCN documents to form a TTCN system. Trying to move a root TTCN document with no children over another (non-collapsed) root TTCN document (possibly with children), will result in that the moved TTCN document will become a child symbol below the other root TTCN document. A non-root TTCN document is moved out of a TTCN system by moving the TTCN document up (or down) one or several times.

However, the order in which "child" icon types to a "parent" icon appear is fixed according to the following:

1. Page icons
2. Association icons
3. Dependency icons
4. SDL Instance icons
5. SDL Dashed icons
6. Child diagram and document icons

The ordering of page icons is specified in the appropriate editor for the diagram/document type (see "SDL Page Order" on page 1784 in chapter 44, *Using the SDL Editor*, or "Diagrams and Pages" on page 1578 in chapter 40, *Using Diagram Editors*). The ordering of association icons, dependency icons, SDL instance icons and SDL dashed icons is fixed and cannot be changed by the user. The ordering of child diagram and document icons can also be changed.

Icon Names and Types

The names of chapters, modules and documents in the Organizer are always shown to the right of the icon.

It is possible to show the *type* of each icon, i.e. a text of the form "System file", "Chapter", "Module", "Object Model", "Block", etc., directly to the right of the icon (ahead of the name). For SDL diagrams, it is also possible to show the virtuality of the diagram. Both these options are available from the menu choice *View Options* in the *View* menu. See "View Options" on page 105 for more information.

If the document is not opened in an editor, the icon text is presented in a normal, plain type face. If the document is opened in an editor, the text is presented in **bold face**.

Double Clicks

To double click on an icon invokes a default action on the information type that the icon represents.

The menu choices and operations corresponding to the double clicks are:

Type of icon	Menu choice / operation
Diagram and document icons	*Edit* on the selected diagram/document itself
Page icons	*Edit* on the selected page in the associated diagram/document
Association and Dependency icons	*Edit* on the diagram the link points to
SDL Instance icons and SDL Dashed icons	*SDL > Type Viewer* The corresponding symbol becomes selected in the Type Viewer. A double-click on the symbol in the type viewer shows and selects the symbol in the parent SDL diagram.
Source Directory and Target Directory icons	*Set Directories*
System File icon	*Configuration > Group File*
Link File icon	*Link > Link Manager*
Header File and Footer File icons	*Edit* on the connected text file
Chapter icon	*Edit* on the chapter, i.e. brings up the *Edit Chapter* dialog for that symbol
Module icon	*Edit* on the module, i.e. brings up the *Edit* dialog for that symbol
More icon	*Show Sub Symbols*
Bookmark icon	*Edit* on the bookmark, i.e. navigate to the place specified by the associated SDT reference or URL.

Menu Bar

This section describes the menu bar of the Organizer Main window and all the available menu choices.

The menu bar contains the following menus:

- *File Menu*
- *Edit Menu*
- *View Menu*
- *Generate Menu*
- *Tools Menu*
- *Bookmarks Menu*
- *Help Menu*.

Available Menu Choices

The following concepts affect the menu choices that are available in the menu bar.

Long and Short Menus

The user can choose between long and short menus with the menu choice *View Options* in the *View* menu. Menu choices only available in long menu mode are presented with the menu choice name within parenthesis in the textual enumeration of menu choices for a menu in the following sections.

License Dependent Menu Choices

The following menu choices are only available if the corresponding tool is available according to the license configuration:

License	Affected menu choices
Code Generator	*Make*
Simulator	*SDL > Simulator UI*
Validator	*SDL > Validator UI*

Configurable Menus

In the SDL suite, some menu choices may be available through the concept of user-defined menus. For more information, see "Defining Menus in the SDL Suite" on page 18 in chapter 1, *User Interface and Basic Operations*.

File Menu

The *File* menu contains the following menu choices. (Menu choices within parenthesis are not available in short menu mode.)

- *New*
- *Open*
- *Save*
- *Save As*
- *Pack Archive*
- *Unpack Archive*
- *Print > All*
- *Print > Selected*
- *Print > Selected and Colored*
- (*Set Directories*)
- *PC Drives*
- (*Compare System*)
- (*Merge System*)
- (*Import SDL*)
- *Recently used system files*
- *Exit*

New

This menu choice displays a dialog, with the following possibilities:

- Start with a new and empty system, containing the basic Organizer view (see "Chapters" on page 46).
- Start with a standard template system, saved as an archive file (*.tgz) in the SDL Suite installation. The archive files for the standard template systems are and should be saved in directory <install dir>/sdt/include/template/.
- Start with any template system, saved as an archive file (*.tgz) anywhere in the file system.

If you want to start with a template system, you will have to specify the directory to unpack the archive file in, in "Unpack Archive" on page 68 that follows.

If a system file already is open in the Organizer, the behavior is determined by the status of the existing system. If modified information exists, the user first gets the possibility to save it; see "The Save Before Dialog" on page 64.

The new system is then created in memory. *Source Directory* and *Target Directory* are left unchanged, i.e. set to the values they had before the *New* operation.

> **Note:**
>
> The actual value of a directory in the *Set Directories* dialog may change if the directory is set to *System file directory*. Since there is no system file associated with a new system, the Telelogic Tau start-up directory is used until the file is saved.

You have to save the system to create a system file on disk.

The old contents of the drawing area is replaced with the basic Organizer view. If any of the documents in the old system managed by the Organizer were opened in an editor, these editor windows are closed.

Open

This menu choice is usually used to open an existing system file. It can also be used to open a single diagram or document file, as well as to open and resolve a *.scu file.

Opening a System File

If a system file is already open in the Organizer, the behavior is determined by the status of the existing system. If the information is not modified, the *Open* dialog is issued (see below). If modified information exists, the user first gets the possibility to save it; see "The Save Before Dialog" on page 64.

The *Open* dialog is a standard file selection dialog, with the file filter set to `*.sdt`. The *Open* button in the dialog opens the specified system file. The old contents of the drawing area is replaced with the new system. If a system window state file is found, see "System Window State File" on page 193, the window position and size is restored to the position and

state it had when the system file was saved. If any of the documents managed by the Organizer were opened in an editor, the editor windows are closed.

If the system file does not specify the *Source Directory* and/or the *Target Directory* explicitly, these directories are set to the directory where the system file was found.

The following information consistency checks are performed when opening a system:

- That the opened file is a valid system file. The case when an SDT-2 diagram file is opened is discussed above. If a system file created with an earlier version of SDT-3 is opened, you will be warned that the file will be saved in the current format
- All file bindings in the system file are verified. For each file, a check is made to see that the file exists and that the file is of the correct type. The file access permissions are also determined. If something is not correct, it is reported in the Organizer Log.

  If a document is marked Invalid, the user must later correct the file binding. The user could either reconnect the document to a valid file, or perform a disconnection in which case the document disappears.
- The SDL diagram structures in the system file and in the SDL diagram files are also compared, in terms of existing SDL reference symbols. If connected SDL diagrams in the system file are not present in the corresponding SDL diagram files, the diagram icons are marked Mismatch and the status is logged in the Organizer Log window. Unconnected SDL diagrams are removed from the system file and the Organizer's diagram structure. New reference symbols found in the SDL diagram files are added as unconnected diagrams in the system file and the Organizer's diagram structure.

  If an SDL diagram is marked mismatched, the user must later correct the diagram structures. The user could either perform a disconnection in which case the diagram disappears, or change the appropriate diagram in an editor to include the mismatched reference symbol.
- File protection of the system file and the working directory. If either of them is write protected, a warning message appears.

Opening a Diagram or Document File

If you are only interested in examining the contents of a single diagram or document file, you can specify the filename of that diagram or document in the open dialog. (It might be helpful to change the filter in the open dialog first, to be able to view the existing files of the type that you are interested in.)

If you specify the filename of a single diagram or document in the open dialog, then a new system file will be created, only containing the specified diagram or document. The diagram or document will also be loaded and shown in an appropriate editor. Note that no sub(structure) diagram files will be visible in the Organizer view.

*Opening a *.scu File*

Software control unit files *.scu are used to allow several people to work in parallel on the same SDL diagram structure. When you update your *.scu files, for instance by using the Configuration > Update menu choice, the SDL diagram structure might change, if someone else has changed the structure and checked in an updated *.scu file.

You can have a top *.scu file associated with the system file symbol in the Organizer drawing area. This *.scu file will take control of the diagram and document structure in the Organizer, leaving the *.sdt file with only control over the user settings (print, view...) and a little system state information (when was the SDL system analyzed without errors last?)

Opening an *.scu file is the same as creating a new system by attaching a *.scu file to the system file symbol in the Organizer and using Configuration > Update to update the diagram and document structure according to the contents of the *.scu file.

Opening an Archive File

Opening an archive file (*.tgz) invokes the unpack archive file operation. Read more about this in "Unpack Archive" on page 68.

Save

This menu choice saves all modified documents and control unit files known to the Organizer, the link file, and finally the system file used by the Organizer. You can still perform a save even if the Organizer contains a completely new system, or if the system has not been changed

since the last save operation. The menu choice has the text *Save (not needed)* in this situation.

Whenever the system file is saved, the system window state file is saved as well. See "System Window State File" on page 193.

If the system file is modified and needs saving, an asterisk '*' is appended to the name of the system file in the Organizer's title bar.

When the first document that is modified is encountered in the Organizer's view of files, the *Save* dialog below appears. If not *Save All*, *Quit All* or *Cancel* is selected, the dialog will remain on the screen and all modified documents will be handled by the dialog subsequently.

Whether the document is connected to a file or not will affect the layout and behavior of the *Save* dialog. For any unsaved and unconnected documents found, the user must provide a filename to connect to.

If the system has been saved before, the system file is saved (without a dialog) after all diagrams and documents have been saved. If the system never has been saved, the Organizer presents a dialog and proposes a name for the system file; the prefix is the name of the first document in the structure, the extension is `.sdt`.

The fields and buttons in the *Save* dialog are:

- *Save in file*

  If the document is connected, the name of the connected file is shown. If the system file is to be saved for the first time, a proposed name for the system file is shown. The filename can be edited by the user.

  If the document is not connected, the Organizer proposes a filename based on the document name and a file extension corresponding to the document type, making the file name unique in the file system (see "Save" on page 11 in chapter 1, *User Interface and Basic Operations* for more information). The filename can be edited by the user.

  The Organizer will not accept a file name that would overwrite another document or a diagram file that is loaded in an editor.

  The user is prompted to confirm the file name if it is already used by a document included in the document structure.

If the file exists in the file system when the *Save* button is pressed, the user is warned in a message box that the existing file will be overwritten.

If a valid filename is provided, *Save* or *Save All* below also connects the diagram to the supplied file.

- *Save*

  Saves the document file. Then the next file which needs to be saved is shown.

- *No Save*

  Ignores the file without saving it. Then the next file which needs to be saved is shown. If, during the save process, the user saves some SDL diagram files but not others, there is a risk of SDL diagram structure mismatches between the system file and the diagram files. Therefore, a warning dialog with the following alternatives is opened:

  – Clicking *OK* **does not** save the file and continues with the next file.

  – Clicking *Cancel* returns to the original Save dialog, making it possible to save the file.

- *Save All*

  Saves all files (diagram and document files, control unit files, the link file, and the system file) without a confirmation by the user, with the exception of unconnected files, which causes the dialog to appear again.

- *Quit All*

  Quits all files (document files, control unit files, the link file, and the system file) without saving. If, during the save process, the user saves some SDL diagram files but not others, there is a risk of SDL diagram structure mismatches between the system file and the dia-

gram files. Therefore, a warning dialog with the following alternatives is opened:

– Clicking *OK* as a response to an *Exit* operation, completes the exit process without saving any more documents / files.

– Clicking *Cancel* aborts the save process (possibly the exit process) completely.

Save As

This menu choice works as the Save menu choice with the following differences:

- *Save As* is always selectable.
- There is always a Save dialog for the system file.
- The *Save As* menu choice is used to save the system file under a new name. If the system file is saved under the old system file name, the user has to confirm this in a dialog.

The *Save Before* Dialog

Some operations in the Organizer need to save information before the actual operation can be performed. The saving is only performed if modified information exists in the system. In these cases a *Save Before* dialog is opened, which is very similar to a normal *Save* dialog. The dialog title is *Save before <command>* and some buttons may behave differently (see Figure 12 on page 70). If not *Save All*, *Quit All* or *Cancel* is selected, the dialog will remain on the screen and all modified files will be handled by the dialog subsequently.

The *Save Before* dialog is opened for the following menu choices:

- *File* menu: *New*, *Open* and *Exit*

  The save process handles modified documents in all chapters. In the case of *Exit*, unsaved diagrams in editors that are not yet in the document structure of the Organizer are also handled.

  If an SDL diagram is modified, has a diagram substructure and some of these SDL diagrams are either opened in an editor or connected to a file, special care must be taken. If the user chooses not to save such an SDL diagram, inconsistencies between the system file and the diagram files may result. The user is warned in a dialog

and may choose to continue, i.e. not to save, or to return to the Save Before dialog. (See "No Save" on page 63 and "Quit All" on page 63.)

- *File* menu: *Compare System* and *Import SDL*

  The save process handles modified documents in all chapters. However, if any modified documents are opened in an editor, only the system file may be saved, not any of the documents.

- *Generate* menu: *Analyze*, *Make* and *SDL Overview*

  The save process handles modified SDL diagrams in all chapters. The buttons *No Save* and *Quit All* are dimmed.

- *Generate* menu: *Convert to PR/MP*

  The save process handles modified SDL diagrams in all chapters. The buttons *No Save* and *Quit All* are dimmed. If the user clicks *Cancel*, the Convert to PR/MP dialog is opened with the GR source diagram toggle dimmed.

Auto Saving

When selecting any of the *Generate* commands *Analyze*, *Make*, *Convert to PR/MP* or *SDL Overview*, the *Save Before* dialog does not appear if the preference AutoSaveBefore is set. However, unconnected and modified documents still require user interaction. If such documents exist, the dialog appears.

Pack Archive

Pack files related to the system loaded in the Organizer into an archive file (*.tgz).

The archive file has the extension *.tgz and is packed using tar and gzip. This means that the archive file can also be unpacked without using the SDL Suite, from the command line, using g(un)zip and tar.

Note:

Packing and unpacking archive files will only work if the SDL Suite can access the external tar and gzip programs, using the preferences mentioned below. tar and gzip usually exist on a UNIX system. For Windows (MSDOS), gnu versions of tar.exe and gzip.exe have been included with the installation. The external tools can be pointed out with the preferences "TarCommand" on page 240 in chapter 3, *The Preference Manager* and "GzipCommand" on page 240 in chapter 3, *The Preference Manager*.

The first Pack Archive Dialog

To use the pack operation, have the files you want to create an archive file for in the Organizer and invoke the operation. The first pack archive dialog will appear, where you can do the following things:

- **Pack all diagrams into archive**. This text field is used to specify where the archive file should be saved, and under what name. As default, the archive file is saved in the target directory, with a name based on the SDL system diagram name, and with a *.tgz extension.
- **Diagram files are relative to**. This text field is used to specify a source directory for files put into the archive. As default, the source directory specified in the Organizer is used. Note that files outside this directory, that should be included in the archive file, are put in a separate directory named *external*.
- **Include top directory in archive**. To avoid mixing files from an archive with other files when unpacking, it is possible to create a top directory for all files in the archive when packing. As default, a top directory is created, with the same name as the SDL system diagram that is packed.
- **Save system file for archive in**. With this option, it is possible to create a new system file for the archive, only referencing files that really are packed with file references updated for the archive. Use this option (instead of packing the original system file) to get a portable archive file. As default, an archive system file is created, in the archive top directory, with a name based on the SDL system diagram that is packed.

The second Pack Archive Dialog

When pressing the Next button in the first pack archive dialog, the second pack archive dialog appears, where file types to include can be decided. The following file types are included as default:

- Archive System File
- SDL Package Diagrams
- Other SDL Diagrams
- MSCs
- UML Diagrams
- Chapter Files
- Other Text Files
- Index Files
- Header and Footer Files

The following file types are not included as default:

- Original System File
- Generic Files
- Simulator Script Files
- *.scu files

The second pack archive dialog also makes it possible to include files from the source or target directories with specific extensions. These extensions should be specified in the **Also include files with the following extensions** text field (a comma separated list of extension names). As default, *.lst files from the source or target directory are included in this way, with the text field text set to *lst*.

The third Pack Archive Dialog

Pressing the Next button in the second pack archive dialog displays the third pack archive dialog, where all files that will be included in the archive are listed. A file in the list is usually presented in the following way:

```
<file system file name> (-> <archive file name> )
```

In the dialog, it is possible to include or exclude individual files.

- To include a file, press the *Add* button and specify the file in the file selection dialog that appears.
- To exclude a file, select it in the list, and press the *Remove* button.

When you are satisfied with the list of files that will be packed, press the *Pack* button to start the pack operation. Information from the pack operation can be found in the Organizer Log.

Unpack Archive

This operation unpacks files in an archive file (*.tgz), and places them in a directory in the file system.

An archive file is created and unpacked with the external tools *tar* and *gzip*, that both must be available for the operation to work. The external tools can be pointed out with the preferences "TarCommand" on page 240 in chapter 3, *The Preference Manager* and "GzipCommand" on page 240 in chapter 3, *The Preference Manager*.

An archive file can be unpacked in one of the following ways:

- By invoking the unpack archive operation in SDL Suite:
  - Select the Organizer > File > Unpack Archive menu choice.
  - Specify a *.tgz file in the Organizer > File > Open dialog.
  - Start SDL Suite with a *.tgz file as a parameter.
- From the command line, using gzip and tar directly. (For more information about this alternative, see the documentation for these tools.

The Unpack Archive Dialog

Whenever SDL Suite is used to unpack, and the unpack directory has not been specified, the Unpack Archive dialog appears. The dialog has three things that should be considered:

- **Unpack archive file**. In this text field, the archive file to unpack should be specified.
- **In directory**. In this text field, the directory to unpack the files in should be specified.
- **Open unpacked system file in Organizer**. If this option is on, the Organizer will after the unpack operation search for a system file (*.sdt) among the unpacked files, and if there is a system file, it will be opened in the Organizer.

Print > All

Similar to *Print > Selected*. The difference is that for *Print All*, the selection is not considered, all diagrams and documents are always printed, if the set of diagrams and documents to print is not further refined in the print dialog.

Print > Selected

Prints all or some of the diagrams in the Organizer. See chapter 5, *Printing Documents and Diagrams*, for more information about the dialog and some examples of how to print. The Organizer selection decides the set of diagrams and documents to print, if the set is not further refined in the print dialog.

Print > Selected and Colored

Similar to *Print > Selected*. The difference is that for *Print Selected and Colored*, only pages (when pages are shown in the Organizer drawing area) or diagrams (when pages are not shown in the Organizer drawing area) with at least one colored symbol are printed. A symbol is colored when the symbol has a border color other than black or a fill color other than white.

Print > Selection File

A file selection dialog appears, where a print selection file should be specified (*.sel). The print selection file is read, and the Organizer print dialog is displayed, updated to reflect the state that the print dialog had when the print selection file was saved from the print dialog. The print selection file is capable of remembering:

- Selected diagrams/documents/pages.
- Print options (scale, paper format...)
- Index Viewer filter options (filter types, uses and diagrams)
- State Matrix Viewer filter options (in Text Editor) (filter processes)

Note that print selection files can also be used by having a print selection symbol in the Organizer. In that case, you can use the print selection file and invoke the print dialog by double-clicking on the print selection symbol.

Set Directories

This menu choice sets the source and target directories. For more information on these directories, see "Source Directory" on page 43 and "Target Directory" on page 43.

This menu choice is only available if the system file can be changed.

If the source directory is changed while a *Save* dialog is active in an editor, the directory where the editor saves the diagram is undefined. An ongoing analysis is not affected by changing the target directory.

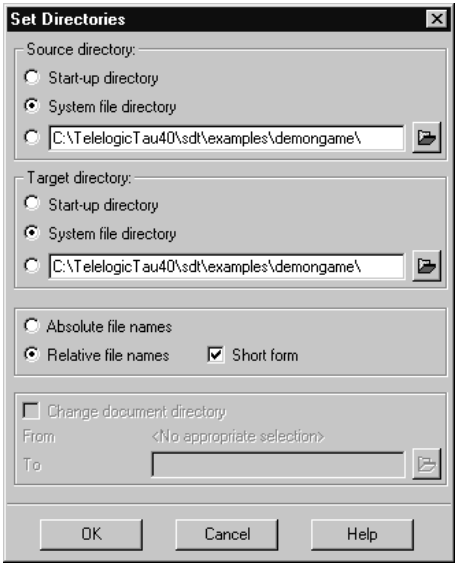

Figure 12: The Set Directories dialog

- *Source Directory*
- *Target Directory*

  Source and target directory settings are saved in the system file. Both these directories can be specified in three ways:

  – SDL suite start-up directory: Source or target directory is set to the directory where the Telelogic Tau tool was started from (**in Windows** the directory of the executable file or the specified start directory for a shortcut icon). This setting means that rela-

tive file bindings in the system will be evaluated starting from the start-up directory the next time the system file is loaded into the Organizer.

– *System file directory:* This is the default value for both source and target directory. Source or target directory is set to the directory where the system file resides. For a new system that has not been saved yet, the start-up directory is used until the system is saved. This setting means that relative file bindings in the system will be evaluated starting from the system file directory the next time the system file is loaded into the Organizer. This setting also makes it possible to move a system file and all related diagram and document files to a new directory without having to update the system file, provided that the positions of all diagram and document files relative to the system file are preserved.

– A specific directory. The directory specified in the text field will be used. Note that the target directory can here be specified using a relative file name. If this is done, the source directory is used as a base to dynamically calculate an absolute target directory. This is useful when you want to have a target directory as a sub directory to the source directory and you want to be able to move all your Telelogic Tau files in or between file systems without great effort.

• *Absolute file names*

  If this option is set, the Organizer shows and stores document files with full (absolute) path.

• *Relative file names*

  If this option is set (the default), the Organizer stores document files with paths relative to the source directory. There are two variants regarding showing document files in the Organizer drawing area:

  – *Short form* on (the default). The file connection is shown exactly as it is stored in the system file.

  – *Short form* off. The directory part of a file connection is shown in *italics* for a file connection that exactly matches the *Source Directory* path. All other file connections are shown as absolute file names, even if they are not necessarily stored with absolute file paths in the system file.

- *Change document directory*

  This operation is used to change the directory part of one or more file connections in one operation. A symbol that has a file connection including the directory that should be changed has to be selected before invoking the dialog. This operation will be dimmed if there is no selection or if the selected file has no file connection.

  In the *To* text field, the directory which the selected directory should be changed into should be specified. When pressing OK with *Change document directory* on, all file connections matching the *From* directory will be changed to the *To* directory.

PC Drives

This menu choice displays the *drive table* of the currently opened system, i.e. the mapping between drive names in Windows and the beginning of corresponding directory paths on UNIX. For more information, see "Windows and UNIX File Compatibility" on page 209.

This menu choice is only available if the system file can be changed.

The following dialog is opened:

Figure 13: The PC Drives dialog

The text area displays the drive table currently used in the system. The table can be edited directly in the text area. When the system file later is saved, the table is stored in the The Drives Section.

When clicking *OK*, a basic syntax check is performed on the entered drive table. Each line should consist of two items only:

- A Windows path, either a drive letter followed by a colon, e.g. `H:`, or a full path (e.g. `C:\TEMP`). UNC paths can be used (e.g.

`\\<host>\file`). If you include a trailing backslash (optional) you must also include a trailing slash in the corresponding UNIX path.

- A UNIX directory path starting with a slash '/'. If you include a trailing slash (optional) you must also include a trailing backslash in the corresponding Windows path.

Paths containing spaces must be put within double quotes. Note that within double quotes, each backslash must be entered as two backslashes, i.e. a UNC path `\\host\dir name` (containing spaces) must be entered as `"\\\\host\\dir name"`.

If any errors are found, the user is notified and the dialog is not closed. If the table was changed and found to be syntactically correct, the system file is marked as modified.

Note:

Changes made in the *PC Drives* dialog only take effect when the system file is reloaded.

Compare System

Works as *Merge System*, except that there is no possibility to merge the differences found during the compare operation.

Merge System

This menu choice compares the contents of the Organizer Main window with the contents of a system file (`.sdt` file). The compare operation is performed on a diagram/document level.

This menu choice is only available if the system file can be changed.

The two systems are compared and possible differences are reported to the user, with the option to merge them, by specifying which documents to add and which to remove in the Organizer system.

The information is processed according to the following scheme:

1. The user is asked to exit any editors, if any is found running. This means that modified documents must be saved before the comparison can be started.
2. If the system file is modified, the user is prompted to save it in a The Save Before Dialog.

3. A standard file selection dialog is issued, where the system file to compare the Organizer system with may be selected. Choosing a system file in this dialog starts the compare operation.

4. If there are differences between the options set in the Organizer and the options saved in the system file to compare with, these are reported first, as a text in a separate dialog. The text might look like this:

```
Comparing system in Organizer
with /home/develop/lat/target/DemonGame.sdt.

Options saved in system file differs:

SemanticControl differs: False True
Kernel differs: SCTVALIDATOR SCTADEBCOM
```

   If you want to remove these differences, you have to do it manually by changing different settings in the Organizer. For instance, to remove the SemanticControl difference by updating the Organizer settings, bring up the Analyzer dialog and select *Semantic analysis*. (*SemanticControl* is the word used in the system file, *Semantic analysis* are the words used in the graphical user interface for the same thing.)

5. The *Compare System* dialog is issued, where the diagram and document differences are reported (if the Organizer system and the contents of a system file are found identical, this is reported in a message box and the operation is terminated).

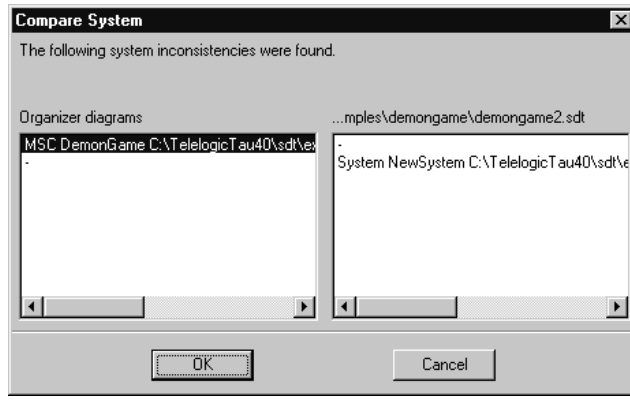

Figure 14: The Compare System dialog

6. The user decides if and how to merge the two views, by selecting or deselecting the items to include or exclude from the resulting system. *OK* updates the system in accordance with the settings.

The *Merge System* Dialog

- The lists are sorted according to the order of appearance of the items in the Organizer Main window and the order of appearance in the system file.
- Each item in a list identifies a chapter, module or document that was found in only one of the two systems (i.e. in the Organizer or in the system file) or were found in both systems, but differ from each other (see "How Systems Are Compared" on page 77 for more information about how structures and documents are compared).

A document is listed with its type, name and the file it is stored on. A module is listed with its type and name. A chapter is listed with its name.

  - A chapter, module or document that was found in one system but missing in the other, is present in one list only and is identified in the other list with a non-selectable '-' (hyphen).
  - Documents that were found in both systems and that are considered *equal* are not listed. See "equal" on page 77 for more information.
  - Documents that were found in both systems and that are considered *almost equal* are presented in both lists. See "almost equal" on page 78 for more information.

- Association and dependency links are not included in the lists, but are preserved as far as possible. See "Associations and Dependencies" on page 78 for more details.
- The items in the two lists that are selected will be included in the resulting (merged) Organizer view.

The items in the *Organizer diagrams* list are the icons that were found in the Organizer's chapters.

By default, all items that originate from the Organizer are selected, meaning that they will be included in the merged view.

Figure 15: A diagram originating from two sources

The MSC DemonGame has been found both in the current Organizer view (where the diagram is connected to a file) and in the system file (where it is connected to another file). The dialog suggests by default to include the diagram originating from the Organizer (the left list) and to exclude the diagram originating from the system file. The user may however change the selection to merge the systems in a different way.

The <*system file*> list indicates the directory and name of the selected system file. The items in this list are the items that were found in the system file.

By default, given two documents that are considered *almost equal*, the dialog will select the document originating from the Organizer, not the document originating from the system file. See example in Figure 15, above.

How Systems Are Compared

Some rules that govern how systems are compared:

- **Relative file names**:

  Relative file names are managed as if they were relative to the *Source Directory* as currently specified in the Organizer.

- Handling of **SDL diagrams** (including packages, excluding macro diagrams):

  SDL diagrams in a tree structure are examined starting from the top and down. When two diagrams are found to differ, their diagram subtree is not examined further, the subtrees being considered as a part of the diagrams that differed.

  This means that if the user chooses to include a diagram that has a subtree, the complete subtree will also be included. Similarly, if the user chooses to exclude a diagram that has a subtree, the complete subtree will be excluded.

  The level of indentation used when listing SDL diagrams in the dialog indicates the structural level at which a diagram is found.

- Handling of **Object Model diagrams**, **State Chart Diagrams, HMSC diagrams, MSC diagrams**, **SDL macro diagrams**, and **SDL overview diagrams**:

  Since these diagrams cannot be part of a subtree, adding or removing any of these does not affect the remaining parts of the structure.

- **Rules for equality of diagrams:**

  Two diagrams/documents with the same type and name can be *equal* or *almost equal*.

  – **equal**

They are considered *equal* if the file names are equal (equal diagrams/documents are not listed in the dialog).

– **almost equal**

They are *almost equal* if the file names are not equal. Two documents that are almost equal are presented on one line in the dialog. For root documents it is possible to select both diagrams on one line in the dialog. For non-root documents it is only possible to select one of the diagrams.

Note:

Compare System and *Merge System* do only compare the structural system information saved in the system file. Compare System and Merge System do **not** compare the document contents, such as page names.

Associations and Dependencies

The Compare System function preserves, as far as possible, the association and dependency links that exist between documents:

1. When an association link is found in a system file, the file names of the two documents are saved.
2. If a document with an association link is selected by the user to be included in the Organizer, a new link is generated if a document with the previously saved file name can be found in the Organizer structure. If such a document cannot be found, the association link is removed from the included document.

Import SDL

Imports an SDL diagram or a number of SDL diagrams in SDT-2 or SDT-3 format, and extracts the diagram structure with the possibility to save it in a system file. The command may, on demand, convert the imported diagrams into SDT-3 format without the need to involve the user for each diagram to convert. It is not possible to import an SDT-3 system file.

This menu choice is only available if the system file can be changed.

To avoid potential name conflicts when saving an imported diagram structure, the menu choice will not perform any action, and causes a message box to be displayed, in the case any files are opened in an editor.

Basically, the involved diagrams and their corresponding files are bound and presented in the Organizer. A number of information entities can be extracted from the diagrams.

If a system file already is open in the Organizer, the behavior is determined by the status of the existing system structure. The user is first asked to exit any editors, if any is found running. This means that modified documents must be saved before the import is started. If a modified system file exists, the user first gets the possibility to save it; see "The Save Before Dialog" on page 64.

The following dialog is then opened:

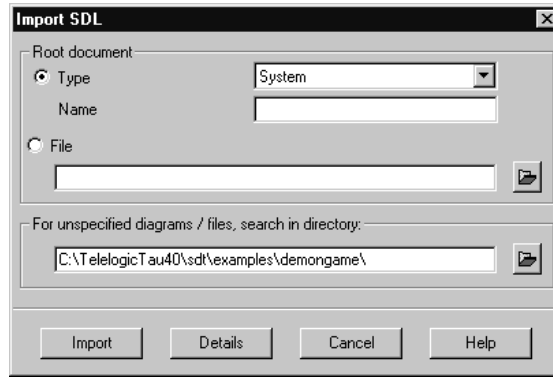

Figure 16: The Import SDL dialog

The root diagram to import and convert can either be named logically with type and name, or physically with a file name. The specified diagram will become a new root diagram, placed last in the Organizer.

- *Type*

  The type of the diagram is selected with this option menu. Used when naming a diagram logically.

- *Name*

  The name of the diagram. Used when naming a diagram logically.

- *File*

  The name of a diagram file. Used when naming a diagram physically.

- *For unspecified diagrams/files, search in directory*

  Specifies that search for diagram files should be done in the specified directory.

- *Import*

  Starts the import operation. Both a root diagram and a search method must be specified in order to start the conversion. If this is not done, an error message is shown. The Organizer log window informs about the progress and result of the conversion.

- *Details*

  Issues this dialog:

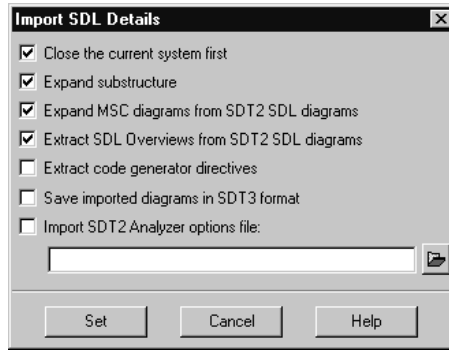

Figure 17: The Import SDL Details dialog

– *Close the current system first*

  If this option is set (the default), the current system in the Organizer will be closed and the SDL diagrams will be imported in a new system. If this option is not set, the SDL diagrams will be imported and added to the current system.

– *Expand Substructure*

  If this option is set (the default), the conversion will be done recursively in the diagram substructure from the specified root diagram. If not set, files corresponding to referenced diagrams are not imported.

– *Extract MSC Diagrams from SDT2 SDL diagrams*

  If this option is set (the default), references to MSC diagrams found in SDL diagrams saved in SDT-2 format will be extracted and placed after the SDL diagram structure, together with a file binding. An association link to the MSC diagram is also inserted in the SDL diagram structure where the MSC diagram was extracted.

– *Extract SDL Overviews from SDT2 SDL diagrams*

  If this option is set (the default), existing overview diagrams stored in SDL diagrams saved in SDT-2 format will be extracted and stored in a separate file. The overview diagram is placed after the SDL diagram structure. An association link to the overview diagram is inserted in the SDL diagram structure where the overview diagram was extracted.

– *Extract code generator directives*

  If this option is set, the diagrams will be searched for a number of directives to be included in the system file. The option is not set as default. The directives #SEPARATE and #WITH are handled

  #SEPARATE: a separator is set in the system structure on each diagram containing the directive (see "Separator symbols" on page 108).

  #WITH: a warning is issued in the Organizer log and a template makefile is generated with the object files found.

  For more information on directives, see "Directives to the Cadvanced/Cbasic SDL to C Compiler" on page 2649 in chapter 57, *The Cadvanced/Cbasic SDL to C Compiler*.

– *Save imported diagrams in SDT 3.X format*

  If this option is set, parts of the diagrams which are obsolete in SDT-3 are removed from the diagram files to convert them to SDT-3 format. The parts removed are the ones controlled by the options *Extract SDL Overviews from SDT2 SDL diagrams* and *Extract MSC Diagrams from SDT2 SDL diagrams*, described above. If these options are not set, some information may be lost when importing SDT-2 diagrams. In this case, the user will be warned in a dialog when pressing the *Import* button:

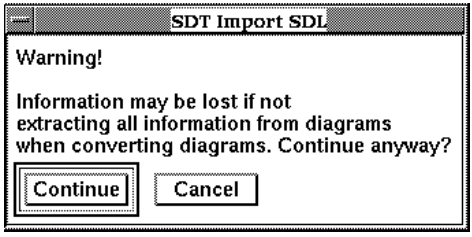

Figure 18: Import diagrams warning

– *Import SDT-2 Analyzer options file*

  If this option is set and a file is specified, the Analyze and Make options found in an SDT-2 Analyzer file (typically `<system>.san`) will be used.

Recently used system files

Just above the *Exit* menu choice, there can be up to four menu choices representing recently opened system files. To open one of them again, select the appropriate menu choice. The information about recently used system files is saved in a file called `.sdtfiles`, in your home directory (see "Environment Variables" on page 47 in chapter 4, *System Setup, in the Installation Guide*).

Exit

This menu choice exits the Organizer.

The exit operation consists of four phases:

1. Handling of modified files managed in the Organizer structure.

   If modified information exists in the current system structure, the user gets the possibility to save it; see "The Save Before Dialog" on page 64. The dialog is opened for the first file (document/system file) that is modified in the Organizer's view of files. The user can then choose how to continue. If the user does not select *Save All*, *Quit All* or *Cancel*, the dialog will remain on the screen and all modified files will be handled by the dialog subsequently.

   The link file and the system file are saved last, if necessary. If the Exit process at a later stage is cancelled, all documents in the editors

are still available, since they are not closed until all modified documents are handled.

2. Confirmation of Exit.

   If no analyze, make, simulation or validation jobs are active, the Organizer exits without user confirmation. If there are such active jobs, or a simulator/validator UI is executing, the user has the possibility to force an exit of these jobs:

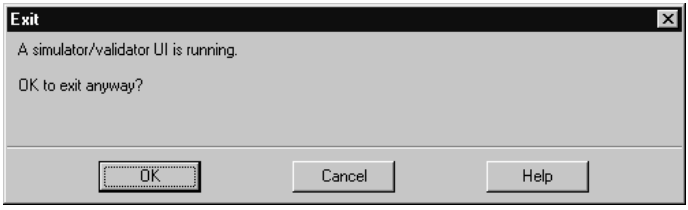

Figure 19: Exit confirmation with active jobs

3. Removing of documents in editors.
4. Shutdown of tools.

   All Telelogic Tau tools are terminated, possibly issuing a *Save before exit* dialog. Finally, the Organizer itself is terminated.

Edit Menu

A general mechanism to edit the document structure(s) in the Organizer does not exist. Some of the menu choices in the *Edit* menu are used for basic operations on root documents and file connections. However, most changes to the document structure are a result of operations made in the diagram editors; see "Reference Symbols" on page 1827 in chapter 44, *Using the SDL Editor*.

The *Edit* menu features the following menu choices (menu choices within parenthesis are not available in short menu mode):

- *Edit*
- *Add New*
- *Add Existing*
- *Remove*
- (*Connect*)

- (*Disconnect*)
- (*Configuration > Group File*)
- (*Configuration > Update*)
- (*Configuration > Full Update*)
- (*Color > Set Default Colors*)
- (*Color > Set Black and White*)
- (*Associate*)
- (*Paste As*)
- (*Go To Source*)
- (*Update Headings*)
- (*Update Visibility*)
- (*Properties*)

Edit

This menu choice edits the selected symbol or document. A document is edited by starting the corresponding editor. A document which is opened in an editor has its name shown in **bold face** in the Organizer.

The menu choice is dimmed if the selected icon is invalid, or if an instance or dashed SDL diagram icon is selected.

The operation performed depends on the type of symbol or document selected, according to the following:

Type of symbol or document	**Operation performed**
System file	The *CM Group* dialog is opened (see "Configuration > Group File" on page 96).
Link file	The Link Manager is opened.
Directory symbol	The *Set Directories* dialog is opened (see "Set Directories" on page 70).

Type of symbol or document	Operation performed
Chapter	The *Edit Chapter* dialog is opened. This dialog contains three choices: • *Edit chapter symbol:* the *Edit* dialog is opened (see below) to allow editing the symbol type, the chapter name, or the connected text file. • *Edit chapter options:* the *Chapter Options* dialog is opened (see "Chapter Options" on page 109). • *Edit first page after chapter:* the first connected SDL diagram or page after the chapter symbol is edited (see below).
Module	The *Edit* dialog is opened (see below) to allow editing the module name.
Connected document or header/footer file	The document/file is opened in an editor. For a diagram, the first page in the diagram is shown, or if the user has specified a page to open first in the editor (see "The Open This Page First Option" on page 1970 in chapter 44, *Using the SDL Editor*), this page is shown instead.
Association or dependency link	The corresponding referenced document is edited as if it was the selected document.
Page symbol	The page is opened in an editor.
Unconnected document or header/footer file	The Edit dialog is opened (see below).

The *Edit* dialog looks like this:

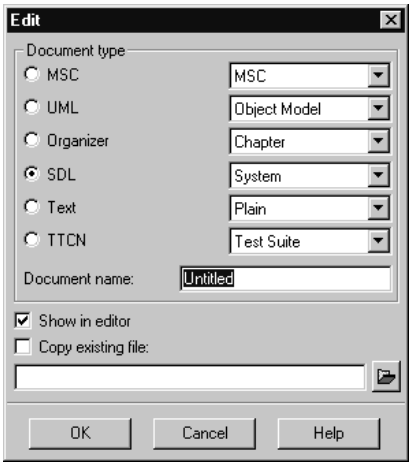

Figure 20: The Edit dialog

- The *Document type* and *Document name* fields can be used to change the type and name of a root document. If a non-root document is selected, these fields are dimmed. If the type is changed, the new document icon will replace the old one. However, the type of a module that contains documents cannot be changed.

  The wanted document type is specified by selecting the correct radio button and selecting the document type in the option menu. It is also possible to select a document in an option menu without selecting the associated radio button first; the radio button will be auto-selected in this case.

- The *Document name* text field contains the current name of the document, chapter or module. The name must conform to the naming rules for the document type; otherwise an error dialog is issued. The name of a module must conform to the naming rules for SDL diagrams, but the name of a chapter may contain any printable characters except quotation marks.

- The *Show in editor* option opens the document in an editor. This option is by default on for document types that can be shown in an editor. Deselecting this option automatically deselects the *Copy existing file* option.

- The *Copy existing file* option copies the specified file and uses it as a starting point/template for the document. If no file is specified, or the file does not exist, no file is copied. Selecting this option automatically selects the *Show in editor* option.

Add New

This menu choice adds a new document to the system. Normally, the added document is placed as a root document below the current selection. If there is no selection, the document is placed as the first root document in the Organizer. Adding a diagram also involves an update of a control unit if there is one that is associated with the diagram substructure affected by the *Add New* command.

There are two exceptions to this:

- If the selection is a module, the new document is instead added at the top level in the module. If the selection is a document in a module, the new document is added in the module at the top level after the selection. To move diagram and documents in and out of modules when they are already in the Organizer, the Move Down and Move Up quick buttons should be used.
- If the selection is a root TTCN document and the user adds a TTCN document, the added document will be placed as a child document below the selected document. If the selection is a child TTCN document, the added document will be placed below the selected document in the same TTCN system. This menu choice is therefore, together with the Move Down and Move Up quick buttons, used to build TTCN systems.

This menu choice is only available if the system file can be changed.

To add SDL diagrams other than root SDL diagrams, the SDL Editor is used (see "Adding a Diagram Reference Symbol" on page 1845 in chapter 44, *Using the SDL Editor*).

The same dialog is opened as for *Edit* on an unconnected document (see Figure 20 on page 87). By default, the *Show in editor* button is on.

If there is a selection, *Document type* and *Document name* in the dialog will be set to the selected symbol's type and name. If there is no selection, the dialog will show the settings from the previous invocation. If

it is the first time this dialog is used, the default type *Module* and the default name “Untitled” will be used.

Multiple root diagrams with the same name are allowed.

If an SDL diagram was selected and an MSC diagram is added, an association link to the MSC diagram is automatically added to the SDL diagram.

Add Existing

This menu choice adds an existing document file to the system. (It is also possible to add several documents by specifying a directory) The existing document is added at the same place as described for the *Add New* menu choice. Adding a diagram also involves an update of a control unit if there is one that is associated with the diagram substructure affected by the *Add Existing* command.

This menu choice is only available if the system file can be changed.

A dialog will be opened, that allows you to specify the file to add, either via a text field or via a standard file selection dialog.

There are also three options in the dialog:

- One to specify if the substructure of the added SDL diagram should be expanded.
- One to specify if the added diagram or document should be shown in an editor.
- One to specify if files existing in sub directories to the specified directory (or the directory where the specified file resides) should be added.

If the standard file selection dialog is used, the file filter is set to reflect the currently selected document type. If a module is selected, the file filter is the same as in the previous invocation. If no document is selected, the file filter is set to `.s??`.

The document type and logical name of the existing document is determined in different ways for different document types. The type and name is determined by:

- Reading the specified file for SDL, MSC, HMSC, OM, SC, and TTCN documents.

- Looking at the file name and extension for text files and generic documents.

> **Note:**
> It is not possible to add a document that does not have a default file extension. For information about default file extensions, see "Save" on page 11 in chapter 1, *User Interface and Basic Operations*.

If an SDL diagram was selected and an MSC or Overview diagram is added, an association link to the MSC or Overview diagram is automatically added to the SDL diagram.

The existing document is by default opened in an editor. This behavior can be changed by the preference Organizer*ShowAddExisting.

Remove

This menu choice removes a selected root document and its document substructure, if any, from the system structure. Modules, chapters and top-level documents in modules can also be removed.

This menu choice is only available if the system file can be changed.

Removing a root document may also involve the update of one control unit file if there is any containing the diagram substructure that has been removed. The menu choice is dimmed if no such document, module or chapter is selected, or if the document is modified.

Removing a chapter symbol does not remove the documents in that chapter; only the chapter symbol itself is removed.

The following dialog is opened:

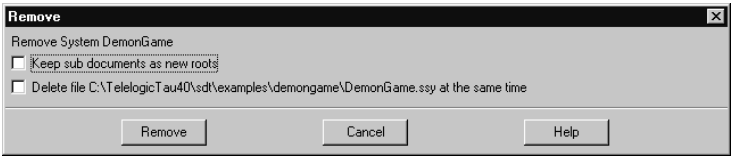

Figure 21: The Remove dialog

- *Keep sub documents as new roots*

If the selected document contains a substructure, this option moves all documents in the substructure to become new root documents (but with their substructures kept intact). It also keeps all bindings to diagrams loaded in an editor. The new root documents are placed directly after the selected root document. If the selected document is a top-level document in a module, the new root documents are placed as top-level documents in the module.

If this option is not set, the document substructure is removed together with the selected document, but documents being edited are still kept as buffers in editors. This is the default setting, except when a module is selected.

The option is dimmed if the document contains no substructure, if no substructure document is connected, or if a chapter symbol is selected.

- *Delete file <file> at the same time*

  This option also deletes the connected file from the file system. It is dimmed if the document is not connected or is opened in an editor, or if a chapter or module is selected.

Connect

This menu choice connects a selected document to a file. It is possible to reconnect an already connected document. The menu choice is hazed if a directory, page, instance diagram, dashed diagram, chapter, or module symbol is selected. The menu choice is also hazed if the loaded system file (*.sdt) or any associated configuration group file (*.scu) file is read-only.

The following dialog appears:

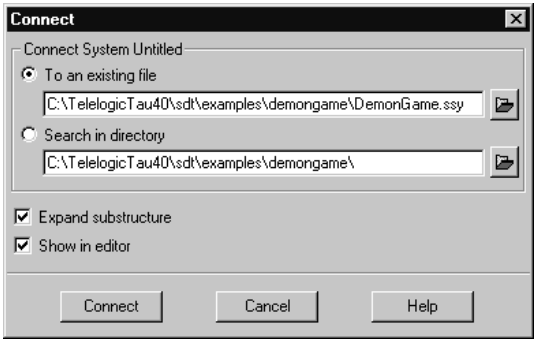

Figure 22: The Connect dialog

- *To an existing file*

  This option connects the document to an existing file. If the document already is connected, the name of the connected file is shown. If the document is unconnected, the field is filled in with the directory component of the file used the last time a document was connected. The filter in the associated file selection dialog corresponds to the file extension for the document type.

  It is possible to use an environment variable to specify the first part of the file path. This can be useful if you want the system to "update its file references" when the environment variable value has changed. For instance, there is an environment variable called "telelogic" that points out the top installation directory for SDL Suite when SDL Suite is run. You can use this environment variable to point out one of the SDL diagram files in the installation:

  $telelogic/sdt/examples/demongame/Main.spr

  (Please consider using relative file names first, because that is often a better solution. See "Set Directories" on page 70.)

  When the connection is to be made, the selected file is inspected. For instance, for an SDL diagram, if the file is not an SDL suite object file, an error message box is issued:

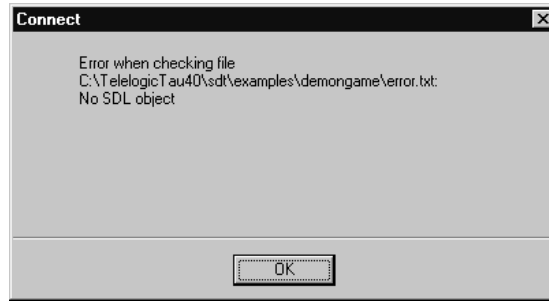

Figure 23: The Connect error message

If the file contains a document of the correct type but with an incorrect name, the symbol in the Organizer is renamed.

If the file contains a document which has an incorrect type or name, the user is warned in a dialog:

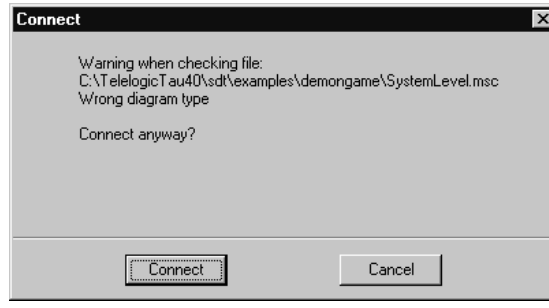

Figure 24: The Connect warning dialog

If the document already is connected and only the directory part of the existing file connection is changed, the following dialog is opened:

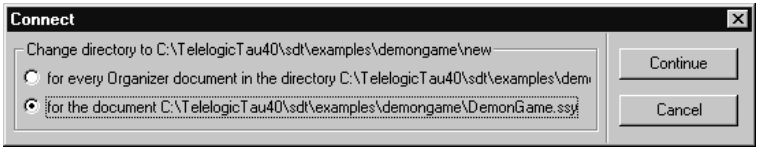

Figure 25: The Connect change directory dialog

- *Search in directory*

  This option searches the specified directory for a document file of a type and name matching the selected document. If such a file is found, the document is connected to the file. If no such file is found, an information box is opened. Pressing *OK* in the dialog returns to the Connect dialog.

  The directory field is filled in with the Source Directory the first time the dialog is used. After that, the directory from the previous usage of the dialog is remembered.

- *Expand substructure*

  This option is only available for SDL diagram symbols. This option recursively expands and connects SDL diagrams to files until no more reference symbols are found. If an SDL diagram has a USE clause, i.e. it references a package diagram, this option also tries to expand the package and put it as a root diagram in the system. The diagram substructure of the package is expanded. Package references are expanded recursively.

  After the expansion is completed, the Organizer display is updated.

- *Show in editor*

  This option opens the document in an editor after the connection has been established.

Error Notification

If an error occurs, the user is informed in a message box and control is returned to the Connect dialog.

Reconnect Connected SDL Diagram

When performing a reconnect to an already connected SDL diagram, the current SDL child diagram references are matched against those found in the connected file. If mismatches are found, icons are marked as such but the structure is kept intact, if possible.

Connect Open Documents

When connecting an unconnected document that is opened and unsaved in an editor, the file name binding is not conveyed to the editor, i.e., the editor binding is lost.

Disconnect

This menu choice disconnects the connected file from the selected diagram. The menu choice is hazed if the selected symbol has no file connection. The menu choice is also hazed if the loaded system file (*.sdt) or any associated configuration group file (*.scu) file is read-only.

The following dialog is opened:

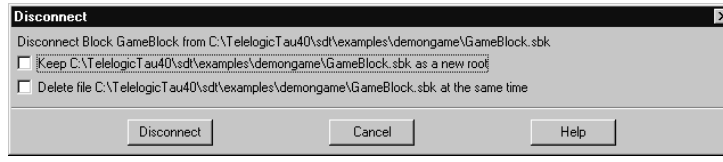

Figure 26: The Disconnect Diagram dialog

- *Keep <file> as a new root*

  This option keeps the connected document in the Organizer structure by making it a new root document. The disconnected document remains in the same place in the Organizer structure.

- *Delete file <file> at the same time*

  This option also deletes the connected file from the file system.

If the document is currently loaded in an editor and is modified, the document reference in the Organizer gets the same status as if a new document is edited, i.e. new and unconnected. The editor binding is then lost.

Configuration > Group File

This command operates on the currently selected diagram, and is used to create or remove a *Configuration Management Group* for the diagram structure. The menu choice is hazed if the loaded system file (*.sdt) or any associated configuration group file (*.scu) file is read-only. When invoked, a dialog appears:

Figure 27: The Group File dialog

- *No group file*

  No group file should be associated with this document; any existing group file for this document group will be removed.

- *Group file*

  Associates a group file with this document, i.e. a *Control Unit* file (*.scu) will be created to hold the structural information about the document and its document substructure. In the text field, the name of the Control Unit file is specified. See "Control Unit File" on page 195 for more information.

  A check is made that the name of the control unit file given by the user is unique within the system. If not, a warning dialogue is issued, making it possible to cancel the operation and to provide a new file name.

  The group file will be shown in the Organizer view like this:

Figure 28: A group file in the Organizer view

The name of the group file is presented directly below the document name, in *italics*. The asterisk '*' indicates that the control unit file is dirty and needs saving. After saving, the asterisk will be removed.

The name of the control unit file is presented directly below the document's file name.

Configuration > Update

A faster version of *Configuration > Full Update*. The *.scu files are only read if they have changed since the last save.

Configuration > Full Update

Updates configuration groups recursively below the selection in the Organizer. Use this menu choice to update the Organizer contents if you have a system with configuration groups loaded in the Organizer and the configuration groups have changed outside the control of the Organizer, for instance by a software configuration management system operation.

Color > Set Default Colors

Invoking this menu choice will make sure that all SDL diagram symbols are colored according to preference values (such as Editor*StateSymbolColor) instead of individual colors set with SDLE > Edit > Symbol Border Color and SDLE > Edit > Symbol Fill Color. Note that this menu choice only operates on the diagrams selected in the Organizer.

Color > Set Black and White

Same as Organizer > Edit > Color > Set Default Colors, but instead of preference values, black and white is used for all symbols.

Associate

This menu choice associates or disassociates a selected document with another document. An association symbol indicates that two document symbols are connected. (A related symbol is the dependency symbol, see "Dependencies" on page 137.)

The menu choice is hazed if the loaded system file (*.sdt) or any associated configuration group file (*.scu) file is read-only.

If an association icon is selected, this menu choice operates on the associated document, not the icon itself. Any document may be associated with any other document, and a document may have more than one associated document.

The following dialog is opened:

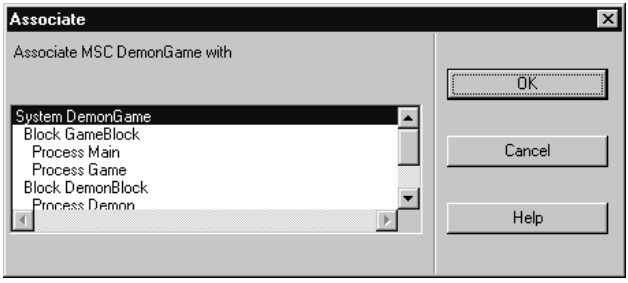

Figure 29: The Associate dialog

- *Associate <document> with*

  The multiple selection list displays the type and name of all documents in the Organizer structure. When the dialog is opened, all documents that the current document is associated with are selected, i.e. the list shows all its associated documents.

  By selecting a new document in the list, an association link to the current document will be created in the selected document's structure. By deselecting a document in the list, the corresponding association link will be removed from the selected document's structure.

Paste As

This menu choice is used to paste copied objects as new diagrams in the Organizer. This menu choice is only available if the system file can be changed. A root diagram is created and opened in an editor. The following transformations are possible via *Paste As* in the Organizer:

- An Object Model class symbol can be pasted as an SDL system diagram.
- An Object Model object symbol can be pasted as an SDL system diagram.
- A text fragment can be pasted as an MSC diagram.

For more information about the Paste As dialog, see "The Paste As Command" on page 448 in chapter 10, *Implinks and Endpoints*.

Go To Source

This menu choice is used to open an editor with a document according to an SDT reference. The SDT reference is specified in a dialog, see Figure 30. If the SDT reference includes information about an object in the document, that object will be selected. SDT references can be obtained by using the menu choice *Show GR Reference* in an editor.

For information about SDT references, see chapter 19, *SDT References*.

The following dialog is opened:

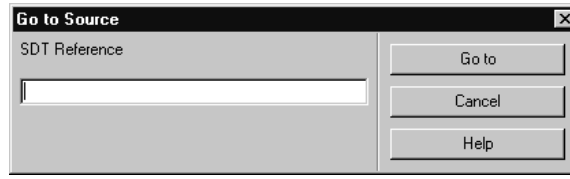

Figure 30: The Go To Source dialog

- *SDT Reference*

  The textual SDT reference is specified in the input field.

- *Go To*

  Shows the symbol in an editor.

An error message appears if the format of the SDT reference is incorrect or if the requested SDT reference cannot be found.

Update Headings

This menu choice checks the headings of SDL, HMSC, OM, and SC diagrams for correctness with respect to what is defined in the Organizer structure.

It operates on the selected diagram and its substructure. If no diagram is selected, all SDL, HMSC, Object Model, and State Chart diagrams in the Organizer are checked. For SDL diagrams, the kernel headings are checked.

Before the headings are checked, a check is made to see if any file is connected to more than one diagram. Such files are reported in the Organizer log, and a warning box is issued to the user. These files may be modified for each appearance and will cause all but the last update to be incorrect.

The heading check is made silently until the first incorrect heading is found. The diagram checked is then shown in the dialog below. If an incorrect qualifier is found, the user is prompted in the dialog whether to update the header or not. The user also has the possibility to silently update all incorrect headings. That is, they are loaded in an editor and are then corrected without confirmation by the user.

After the operation, all updated headings are in an unsaved mode in the editor.

This operation should be done regularly in order to avoid peculiar and hard-to-find analysis error caused by incorrect diagram headings.

In SDL diagrams, qualifiers can be placed in other symbols than the heading in the system, such as qualifying data types in a text symbol. Such qualifiers are not found by the *Update Headings* menu choice.

The following dialog is opened:

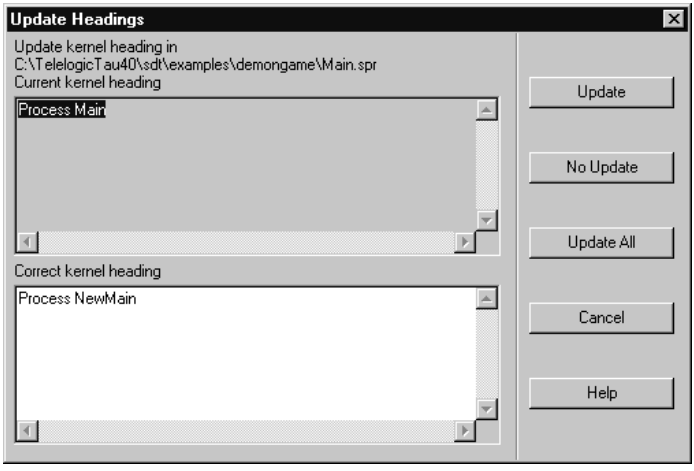

Figure 31: The Update Headings dialog

- *Current kernel heading*

  A read-only text field that shows the contents of the heading as defined in the SDL, HMSC, Object Model, or State Chart diagram, i.e. what is displayed in the editor. (*The Kernel Heading* in SDL diagrams.)

- Correct kernel heading

  A text field with the correct heading according to the structural location of the diagram. This field may be edited by the user.

- *Update*

  Updates the heading in the editor according to the text in the *Correct kernel heading* text field. Continues to search for the next incorrect heading and keeps the dialog open. When the updating is completed, a message appears in the status bar.

- *No Update*

  Does not update the diagram's heading. Continues to search for the next incorrect heading and keeps the dialog open.

- *Update All*

  Updates all headings without a confirmation by the user and closes the dialog. When the updating is completed, a confirmation message is shown in the status bar.

Update Visibility

Update SDL symbol visibility according to include expressions and external synonym values.

The SDL symbol visibility can also be set manually, see "Symbol Visibility > Hide" on page 1948 in chapter 44, *Using the SDL Editor* and "Symbol Visibility > Show" on page 1948 in chapter 44, *Using the SDL Editor*.

For more information about include expressions, see "Include Expression" on page 1948 in chapter 44, *Using the SDL Editor*.

External synonyms are saved in a plain text file with the extension *.syn in the Organizer. To specify that the boolean external synonym variable DEBUG should have the value of true, and the boolean external syn-

onym variable VERSIONTWO should have the value of false, the *.syn file should have the following contents:

```
DEBUG 1
VERSIONTWO 0
```

By using these variable names as include expressions on selected symbols, and by having a reference to the *.syn file in the Organizer, it is possible to hide or show groups of symbols by setting up the correct values in the external synonym file and applying this command. Note that the symbol visibility can not be updated in read-only diagrams.

Properties

Edit bookmark properties for a selected bookmark symbol. *Location* should be set to a valid SDT reference or URL. *Name* is any name that makes it easy to remember the place the bookmark represents. You can get valid SDT references from SDL Suite editors, by selecting a symbol and using <editor>>*Tools*>*Show GR Reference*. The easiest way to create a new bookmark with a valid SDT reference is to use <editor>>*Tools*>*Create Bookmark*.

This menu choice is only available if the system file can be changed.

View Menu

The *View* menu includes the following menu choices (menu choices within parenthesis are not available in short menu mode):

- *Expand*
- *Expand Substructure*
- *Collapse*
- (*Show Sub Symbols*)
- (*Hide*)
- *View Options*
- *Chapter Options*
- (*Set Scale*)
- (*Show High-Level View*)
- (*Show Detailed View*)

Expand

This menu choice expands the symbol structure tree one level down for the selected document. If any symbols one level down are hidden, they

are still hidden after this operation. (Use the menu choice *Show Sub Symbols* to show hidden symbols.)

The menu choice is dimmed if:

- No document is selected
- The selected icon is a leaf (no child icons)
- The selected diagram is not connected
- The selected icon is marked invalid
- The selected icon is already expanded

Expand Substructure

This menu choice expands the symbol structure tree the whole way down for the selected document. This also expands sub symbols that are hidden, but those sub symbols are still hidden after this operation. (Use the menu choice *Show Sub Symbols* to show hidden symbols.)

The menu choice is dimmed if:

- The selected icon is a leaf (no child icons)
- The selected diagram is not connected
- The selected icon is marked invalid
- The selected icon is already expanded

If no document is selected, all icons will be expanded.

Collapse

This menu choice collapses the selected document, i.e. the sub symbols are not shown after this operation. A collapsed document has a small triangle drawn below the icon to indicate that it is collapsed.

The menu choice is dimmed if:

- The selected icon is a leaf (no children icons)
- The selected document is not connected
- The selected icon is marked invalid
- The selected icon is already collapsed

If no document is selected, all icons will be collapsed. A collapsed document does not affect a corresponding document file opened in an editor, i.e. it does not have to be closed or saved.

Show Sub Symbols

This menu choice is used to specify which sub symbols of the selected document that should be shown or hidden. The sub symbols can be documents, instance diagrams, pages, or associations. If a More symbol is selected, the operation applies to the parent document, which becomes selected instead.

Only the sub symbols one level down from the selected document is affected, not the complete symbol substructure. The menu choice is dimmed if there is no selection or the selected symbol has no sub symbols.

Note:

This menu choice does not expand or collapse the document.

The following dialog is opened:

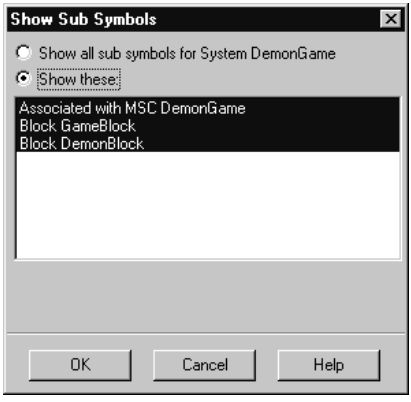

Figure 32: The Show Sub Symbols dialog

- *Show all sub symbols for <document>*

  This option shows all sub symbols for the selected document. It is a shortcut for selecting all sub symbols in the list below.

- *Show these*

  This option shows the sub symbols that are selected in the list, and hides the sub symbols that are unselected. Already shown sub sym-

bols are pre-selected when the dialog is opened. The list is a multiple selection list that toggles the state of the selected item.

Hide

This menu choice hides the selected non-root document and its substructure. The document and its substructure is replaced by a More symbol, which is always placed last of the symbols on that level. If such a symbol already existed in the parent document, the document is hidden under the same More symbol. The symbol's count of hidden documents is updated.

The menu choice is dimmed if there is no selection, or the selected document is a root document.

By double-clicking on the More symbol, the *Show Sub Symbols* dialog is opened.

View Options

This menu choice sets options for controlling the appearance of the Organizer window, as well as options for which icon attributes to show.

The options are set in a modeless dialog, i.e. the Organizer can continue working without waiting for the dialog to be closed. The options are saved in the system file.

The following dialog is opened:

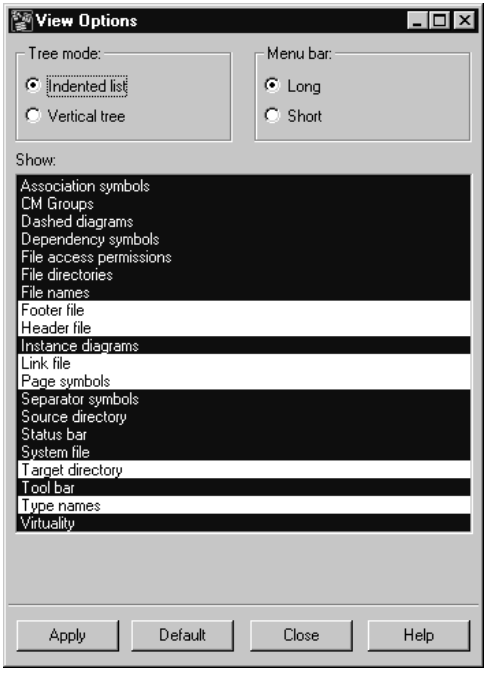

Figure 33: The View Options dialog

The figure above shows the default settings. The settings made in the dialog are preserved as default values the next time the dialog is invoked.

Tree mode

The *Tree mode* section contains options for the two different tree presentation modes available in the chapters (see "Presentation Modes" on page 45).

- *Indented list*
- *Vertical tree*

Menu bar

The *Menu bar* section contains options for which menu choices that are available (see "Long and Short Menus" on page 57).

- *Long*

- *Short*

Show

The *Show* section contains options for which window parts, documents and file attributes to show. The options are available as items in a multiple selection list, which can be selected or deselected. Options already turned on are pre-selected when the dialog is opened.

- *Association symbols*

  Show/hide association symbols (links to associated documents).

- *CM Groups*

  This option governs whether CM Groups should be displayed or not. By default, CM Groups are visible. However, note that they are not shown in vertical tree mode (see "Presentation Modes" on page 45).

- *Dashed diagrams*

  Show/hide dashed diagrams.

- *Dependency symbols*

  Show/hide dependency symbols (links to documents which a document is depending on).

- *File access permissions*

  Add/remove the file access permissions for connected files. An access permission can have one of the following values:

  `rw` Readable and writable
  `r-` Only readable
  `-w` Only writable
  `--` Neither readable nor writable

  In list mode, the permissions are added in front of the file names, if they are shown. In tree mode, the permissions are added on a separate line under each node in the document structure tree, just above the file names if they are shown.

- *File directories*

  Complements the *File names* option below. Add/remove the directory path of the connected files, in the same location as *File names*. If the *File names* option is set, the directory path is added in front of

the file name. The path added is determined by the *Source Directory* option; see "Set Directories" on page 70.

- *File names*

  In list mode: Add/remove a column of connected file names to the documents.

  In tree mode: Add/remove a line of text with the connected filename under each node in the document structure tree. The texts do not overlap; document symbols are separated to make space for the full text.

  If a document is not connected to a file, `[unconnected]` is shown.

- *Footer file*

  Show/hide the Footer File icon.

- *Header file*

  Show/hide the Header File icon.

- *Instance diagrams*

  Show/hide instance diagrams.

- *Link file*

  This option sets the visibility of the Link File icon. The link file icon is hidden by default.

- *Page symbols*

  Show/hide page symbols below SDL, HMSC, OM, and SC diagrams.

- *Separator symbols*

  Show/hide separators between diagrams. Separator symbols show how the generated code will be separated into different files. The lines between diagrams are broken by two vertical or horizontal bars where the separations occur.

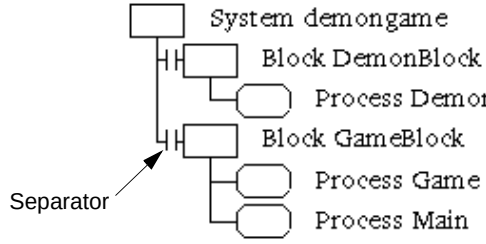

Figure 34: Separators between SDL diagrams

- *Source directory*

  Show/hide the Source Directory.

- *Status bar*

  Show/hide the status bar of the Organizer Main window.

- *System file*

  This option sets the visibility of the System File icon. The system file icon is visible by default.

- *Target directory*

  Show/hide the Target Directory.

- *Tool bar*

  Show/hide the tool bar of the Organizer Main window.

- *Type names*

  Show/hide the type of all icons and documents in textual form, e.g. Object Model, Module, Block, System file, Chapter, etc. The text is placed to the left of the name of the icon.

- *Virtuality*

  Show/hide the virtuality of SDL type diagrams in textual form.

Chapter Options

This menu choice is used to set chapter properties.

The following dialog appears:

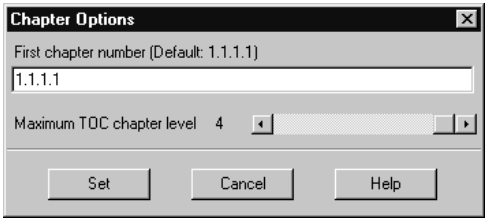

Figure 35: The Chapter Options dialog

The first chapter number is defined in the text field. This number defines the chapter number to use for the first chapter symbol of type Chapter 1, Chapter 1.1, Chapter 1.1.1 or Chapter 1.1.1.1 in the Organizer View area. For example, if the first chapter number is specified as "3.2", then an initial chapter symbol of type Chapter 1 will get chapter number 3. If the initial chapter symbol instead is of type Chapter 1.1.1.1, it will get a chapter number of 3.2.1.1.

The *maximum TOC chapter level* decides which chapter symbols that will be visible in the table of contents when a print is done from the Organizer. A maximum TOC chapter level of zero will only show chapter symbols of type Chapter, while a chapter level of 4 will show all chapter symbol types in the table of contents.

Set Scale

Sets the scale (20%–800%) used in the Organizer window. The setting is saved in the system file.

Show High-Level View

Sets the current view of all diagrams in the SDL Editor to the high-level view, showing only symbols which are marked as "important".

Show Detailed View

Sets the current view of all diagrams in the SDL Editor to the detailed view, showing all symbols.

Generate Menu

The *Generate* menu contains the following menu choices (menu choices within parenthesis are not available in short menu mode):

- *Analyze*
- (*Make*)
- *Stop Analyze/Make (UNIX only)*
- *Targeting Expert*
- *SDL Overview*
- *State Overview*
- (*CPP2SDL Options*)
- (*Convert to PR/MP*)
- (*Convert to GR*)
- (*Convert GR to CIF*)
- (*Convert CIF to GR*)
- (*Convert State Chart to SDL*)
- (*Edit Separation*)
- (*Dependencies*)
- (*Merge ASN.1*)

Analyze

This menu choice analyzes the selected SDL or TTCN system. If there is no SDL or TTCN system selected, the Organizer operates on the first SDL system found in the Organizer. The menu choice is dimmed if:

- A job using the Analyzer is already running.
- No SDL diagram is present in the Organizer and no TTCN system is selected.
- The selected SDL diagram or TTCN document is not connected.
- The selected icon is marked invalid.

The Analyze SDL variant of this menu choice is described below, followed by the Analyze TTCN variant.

Analyze SDL

This menu choice starts the Analyzer for one or several related SDL diagrams. If modified information exists in the current system structure, the user should first save it. See "The Save Before Dialog" on page 64 for more information.

Any SDL diagram can be selected for analysis, but in practice at least the parent block diagram will be the source of the analysis. If no SDL diagram is selected, the first SDL diagram found in the Organizer view will be used. The diagram will be analyzed in its context and together with its substructure.

Options for the Analyzer are specified in the modal dialog below. The settings are saved in the system file and persist until the next time the dialog is invoked for the same system.

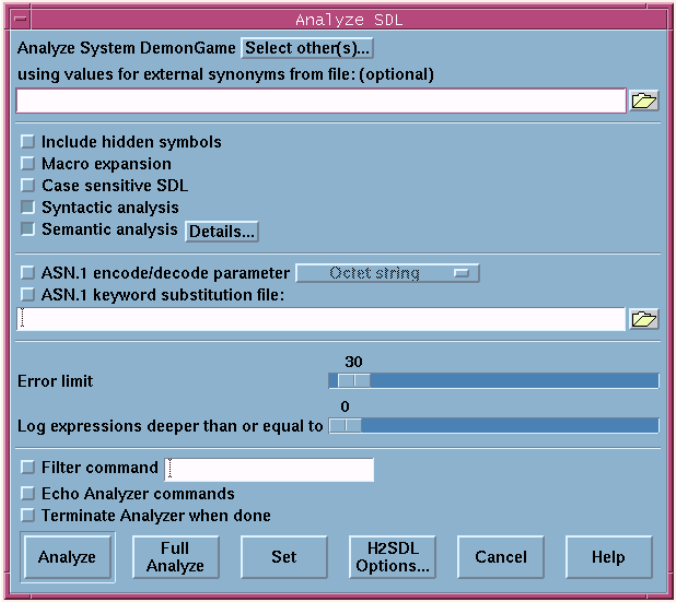

Figure 36: The Analyze dialog

The *Analyze SDL* Settings

> **Note: Normal versus full analyze (make)**
>
> In the *Analyze* and the *Make* dialog, the user can choose between the normal *Analyze*/*Make* and the *Full Analyze*/*Make* buttons. The last used button (normal or full) will be the default when any of these two dialogs is used the next time. The last used button will also determine if normal or full analyze/make will be used when invoking operations via the quick buttons *Analyze*, *Make*, *Simulate* and *Validate*. It is possible to toggle between normal or full analyze/make by pressing `<Ctrl+T>`.

- *Select other(s)*

  When the Analyze dialog is opened, the selection in the Organizer decides the SDL diagrams that will be analyzed. The part that will be considered for analysis is the diagram itself, all diagrams below the diagram and all parent diagrams. A used package is considered to be a parent diagram. All diagrams below a used package will be analyzed.

  This button opens a sub dialog, with the possibility to analyze SDL diagrams other than the one selected in the Organizer. For instance, it is possible to analyze two out of many blocks from the same level in one SDL system. In the sub dialog, just select the diagrams that you want analyzed. Note that you only have to select the top diagram you want analyzed, all sub diagrams will be included in the analysis. The initial label in the Analyze dialog (*Analyze <diagram type> <diagram name>*) will change to reflect the new selections you have made.

- *Using values for external synonyms from file*

  Instead of having a text symbol connected to an external synonym file (*.syn), the external synonym file can be specified here. If there is at least one synonym file in the Organizer view, the one closest to the SDL system will be used as the default value. See "Supplying Values of External Synonyms" on page 2172 in chapter 51, *Simulating a System* for more information.

- *Include hidden symbols*

Decides if hidden SDL symbols should be included in SDL/PR or not. Read more about hidden SDL symbols in *Symbol Visibility > Hide* and *Symbol Visibility > Show*.

If hidden symbols are excluded, then lines to and from hidden symbols will also be excluded. One exception to this is that flow lines going to a symbol that will be excluded, are reconnected to the first following symbol that will be included, if there is one and only one such symbol.

To use include expressions is another way to decide if SDL/GR symbols should be included in SDL/PR or not. Read more about include expressions in *Include Expression*.

- *Macro expansion*

  Run the Macro Expander before the analysis. (See "The Macro Expander" on page 2432 for more information.)

- *Case sensitive SDL*

  Use case sensitive SDL names. If this option is selected, keywords must be all upper or lower case.

- *Syntactic analysis*

  Perform a syntax check.

- *Semantic analysis*

  Perform a semantic check. This option automatically sets the option *Syntactic analysis*.

  Pressing the Details button displays the Semantic Analysis - Details dialog. These options are only used if a semantic analysis is done. They are also not used if a system or package was not the target for the analysis. For more information on these options, see "Perform-

ing Semantic Check" on page 2550 in chapter 56, *Analyzing a System*.

- *Check output semantics*
- *Check unused definitions*
- *Check optional parameters*
- *Check trailing parameters*
- *Check references*
- *Check missing else answers*
- *Check missing answer values*
- *Allow implicit type conversion*
- *Include optional fields in make operator*
- *Generate a cross reference file*

  Generate a cross reference file when performing the analysis. In the text field, a file name is proposed with the diagram to be analyzed as prefix and `.xrf` as extension. The file is by default generated in the Target Directory.

- *Generate a complexity measurement file*

  Generate a complexity measurement file when performing the analysis. In the text field, a file name is proposed with the diagram to be analyzed as prefix and `.csv` as extension. The file is by default generated in the Target Directory. See chapter 49, *Complexity Measurements* for more information.

- *Generate an instance information file*

  Generate an instance information file when performing the analysis. In the text field, a file name is proposed with the diagram to be analyzed as prefix and `.ins` as extension. The file is by default generated in the Target Directory.

- *ASN.1 encode/decode parameter*

  This option decides if the ASN.1 encode/decode buffer can be accessed from SDL. If the option is on, it is also possible to specify how the ASN.1 encode/decode buffer should be represented in SDL. For more information, see "SDL Encoding and Decoding Interfaces" on page 2768 in chapter 59, *ASN.1 Encoding and De-coding in the SDL Suite*.

- *ASN.1 keyword substitution file*

Change keywords in files output by asn1util according to file. If no file is specified, the keywords in the file `asn1util_kwd.txt` in the SDL Suite installation are used. For more information, see "Keywords substitution" on page 701 in chapter 14, *The ASN.1 Utilities*.

- *Error limit*

  The error limit before aborting the analysis (0–1000). A limit of 0 (zero) means that there is no error limit.

- *Log expressions deeper than*

  The performance of the Analyzer depends to a large extent on the depth of the SDL expressions to be resolved. This option is used to emit warnings when the depth of an expression exceeds a certain value (0–100). A limit of 0 (zero) means that no warnings will be issued.

- *Filter command*

  If used, this option allows preprocessing of files before the different analyzer phases. An executable (script), possibly with parameters, should be specified in the text field. The executable will be called before the analyzer processes any file in any phase. The executable will be called with two parameters:

  – The first parameter is the file that is going to be processed.

  – The second parameter identifies the analyzer phase that is going to be executed. Three different phases are distinguished: *import* (before conversion to PR), *macro* (before macro expansion) and *parse* (before syntax analysis).

- *Echo Analyzer commands*

  Print (echo) all Analyzer commands in the Organizer Log as they are executed.

- *Terminate Analyzer when done*

  Terminate the Analyzer process after analysis is done. By default, the Analyzer process is left running in the background after completed analysis.

The options above are forwarded to the Analyzer when the analysis process starts.

The *Analyze SDL* Buttons

- *Analyze*

  Starts the Analyzer in the background and closes the dialog. Status information from the ongoing analysis is shown in the Organizer Log window. Among those diagrams that are considered for analysis according to the top of the dialog (Select other(s)), only the following diagrams are really analyzed:

  – Diagrams that had errors the last time the system was analyzed.

  – Changed diagrams.

  – Diagrams that any diagram mentioned above are depending on.

- *Full Analyze*

  Has the same effect as the *Analyze* button, except that it forces all diagrams that are considered for analysis to be analyzed.

- *Set*

  Saves the Analyzer option settings and closes the dialog, but does not perform an analysis.

Analyze TTCN

This menu choice is used to analyze a TTCN system and/or to generate a Flat View for a TTCN system.

Options for the Analyze process are specified in the modal dialog below. A selectable *Generate Flat View* phase is executed in the analyze process. The settings are saved in the system file and persist until the next time this dialog is invoked for the same TTCN system.

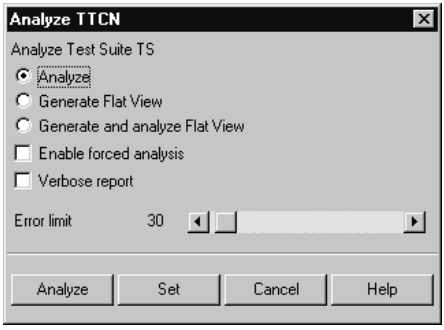

Figure 37: The Analyze TTCN dialog

The *Analyze TTCN* Settings

- *Analyze*

  Selecting this radio button means that the complete TTCN document will be considered for analysis. This radio button is set as default when the dialog is invoked.

- *Generate Flat View*

  Selecting this radio button means that a Flat View will be generated for the selected TTCN system. For a detailed description of this operation, see "Generating a Flat View" on page 1192 in chapter 27, *Analyzing TTCN Documents (on UNIX)*.

- *Generate and analyze Flat View*

  Selecting this radio button means that a Flat View will be generated for the selected TTCN system and it will be considered for analysis.

- *Enable forced analysis*

  This option is described in "Enable Forced Analysis" on page 1185 in chapter 27, *Analyzing TTCN Documents (on UNIX)*.

- *Verbose report*

  This option is described in "Verbosity" on page 1185 in chapter 27, *Analyzing TTCN Documents (on UNIX)*.

- *Error limit*

The error limit before aborting the analysis (0–1000). A limit of 0 (zero) means that there is no error limit.

The *Analyze TTCN* Buttons

- *Analyze*

  Starts the Analyzer and closes the dialog. Status information from the ongoing analysis is shown in the ITEX log. For more information, see "The TTCN Suite Logs" on page 6 in chapter 1, *User Interface and Basic Operations*.

- *Set*

  Saves the Analyzer option settings and closes the dialog, but does not perform an analysis.

For more information about this dialog, see "The TTCN Analyzer" on page 1182 in chapter 27, *Analyzing TTCN Documents (on UNIX)*.

See also chapter 32, *Analyzing TTCN Documents (in Windows)*.

Make

This menu choice makes the selected SDL or TTCN system, or the selected SDL block or process diagram. If a Build Script containing commands to the SDL to C Compiler is selected, that file will be used as input to the SDL to C Compiler, without opening the *Make* dialog (see "Build Scripts" on page 2572 in chapter 57, *The Cadvanced/Cbasic SDL to C Compiler*.

If no document or file of the above mentioned type is selected, the Organizer operates on the first SDL system found in the Organizer. The menu choice is dimmed if:

- A job using the Analyzer is already running
- There is no SDL system diagram in the chapters and no TTCN system is selected
- The selected diagram is not connected
- The selected icon is marked invalid

If modified information exists in the current system structure, the user should first save it; see "The Save Before Dialog" on page 64.

The *SDL Make* variant of this menu choice is described below, followed by the *TTCN Make* variant.

SDL Make

Options for the Make process are specified in the modal dialog below. An analysis phase is executed as part of the Make process. The existing Analyzer options as set in the *Analyze* dialog are used.

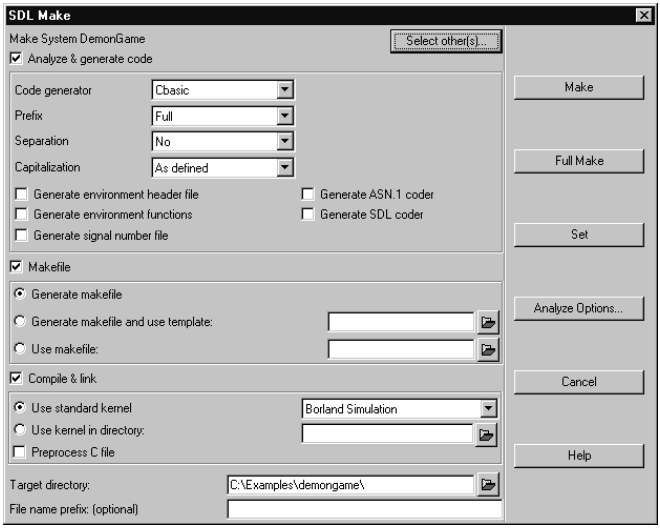

Figure 38: The Make dialog

Code Generation Options

- *Select other(s)*

  When the Make dialog is opened, the selection in the Organizer decides the diagrams that will be used when code is created. This button opens a sub dialog, with the possibility to generate code for diagrams other than those selected in the Organizer. For instance, it is possible to generate code for two out of many blocks on the same level in one SDL system. In the sub dialog, select the diagrams that you want to include in the code generation process. Note that you only have to select top diagrams, sub diagrams will be automatically included. The initial label in the Make dialog (*Make <diagram type> <diagram name>*) will change to reflect the new selections you have made.

- *Analyze & generate code*

This option generates code, and performs an analysis if necessary. If not set, the settings below do not affect the Make process.

- *Code generator*

  The code generator is selected with this option menu. The following code generators are supported to date:

  – *Cbasic* (see chapter 57, *The Cadvanced/Cbasic SDL to C Compiler*)
  – *Cadvanced* (see chapter 57, *The Cadvanced/Cbasic SDL to C Compiler* and chapter 62, *The Master Library*)
  – *X*[1] **(UNIX only)**

- *Prefix*

  The type of prefix for variables is selected with this option menu: *Full* (default), *Entity Class*, *No* or *Special*. For more information, see "Prefixes" on page 2664.

- *Separation*

  The type of modularity is selected with this option menu: *No* (default), *Full* or *User Defined*. For more information, see "Selecting File Structure for Generated Code – Directive #SEPARATE" on page 2650.

- *Capitalization*

  The type of capitalization is selected with this option menu: *Lower Case* or *As Defined* (default). For more information, see "Case Sensitivity" on page 2666.

- *Generate environment header file*

  If this option is set, a header file is generated containing the definitions of the SDL system's interface to the environment. For more information, see "System Interface Header File" on page 2704.

- *Generate environment functions*

  If this option is set, environment functions are generated. For more information, see "The Environment Functions" on page 2702.

- *Generate signal number file*

1. X is reserved for future extensions in the SDL suite applied to code generation.

If this option is set, a file with signal numbers will be generated. For more information, see "Generation of Support Files" on page 2575.

- *Generate ASN.1 coder*

  If this option is set, encoders and decoders from ASN.1 modules will be generated. This option is only available through a special license.

- *Generate SDL coder*

  If this option is set, encoders and decoders from SDL will be generated, see "Type description nodes for SDL types" on page 2718. This option is only available through a special license.

Makefile Options

- *Makefile*

  This option controls makefile creation/usage. If not set, a makefile will not be created/used.

- *Generate makefile*

  Generate a makefile (default). Choose the file by adding a `.m` extension to the separation name of the selected unit and the file is created in the directory specified as target directory. See "Target Directory" on page 70 for more information.

- *Generate makefile and use template*

  Generates a makefile and appends the specified, user defined, template at the end of the makefile. Two "hooks" are provided in the generated part of the makefile: USERTARGET and USERLIBRARIES. This enables the user to define his own targets as well as adding properties in the make file. The recommended file name extension for a template file is `.tpm`.

  In the template file, USERTARGET is used to add additional object files to the link script in the generated make file, by defining this name as a list of object files.

  In the template file, USERLIBRARIES is used to add library modules, for example `-lm` or `-lsocket`, to the link script in the generated make file, by defining this name as a list of libraries.

The template file can also contain the compilation scripts for the object files specified as USERTARGET.

Example 2: Contents of a UNIX Make Template File ––––––––––––––

```
USERTARGET = sctenv$(sctOEXTENSION)
USERLIBRARIES = -lm -lsocket

# Dependencies and actions
sctenv$(sctOEXTENSION): sctenv.c
        $(sctCC) $(sctCPPFLAGS) $(sctCCFLAGS) \
        $(TARGETDIRECTORY) sctenv.c \
        $(sctIFDEF) -o sctenv$(sctOEXTENSION)
```

––

- *Use makefile*

  Specifies an existing makefile to use.

Compile & Link Options

- *Compile & link*

  This option controls if compilation and linking should take place. If set, compilation or linking will be done according to the settings below.

- *Standard kernel*

  Use one of the available standard kernels. The kernel is selected from this option menu. The available kernels depend on the license configuration.

  Each kernel is available in versions for different compilers. You should use a kernel corresponding to a compiler you have access to on your system.

On UNIX, the following kernels are available. There are versions for standard ANSI C compilers (e.g. cc) and for the GNU C compiler (gcc).

- *Simulation*
- *gcc-Simulation*
- *RealTimeSimulation*
- *gcc-RealTimeSimulation*
- *PerformanceSimulation*
- *gcc-PerformanceSimulation*
- *Application*
- *gcc-Application*
- *ApplicationDebug*
- *gcc-ApplicationDebug*
- *Validation*
- *gcc-Validation*
- *TTCN-Link*
- *gcc-TTCN-Link*

In Windows, the following kernels are available. There are versions for the Borland C++ compiler (bcc32) and the Microsoft Visual C++ compiler (cl).

- *Borland Simulation*
- *Microsoft Simulation*
- *Borland RealTimeSimulation*
- *Microsoft RealTimeSimulation*
- *Borland PerformanceSimulation*
- *Microsoft PerformanceSimulation*
- *Borland Application*
- *Microsoft Application*
- *Borland ApplicationDebug*
- *Microsoft ApplicationDebug*
- *Borland Validation*

Information about the kernels can be found in "Compilation Switches" on page 3041 in chapter 62, *The Master Library* and "Libraries" on page 2698 in chapter 58, *Building an Application*[1]. A list of the available kernels can also be found in the file `sdtsct.knl` (see "File sdtsct.knl" on page 3063 in chapter 62, *The Master Library*).

1. The TTCN-Link kernel is described in the ITEX manuals.

- *Use kernel in directory*

  Specifies a non-standard kernel to use. The user should specify the directory where the actual kernel is stored.

- *Preprocess C file*

  This option informs the SDL to C Compilers if the SDL C Compiler Driver (SCCD) should be invoked. See chapter 61, *SDL C Compiler Driver (SCCD)* for more information.

Miscellaneous Options

- *Target directory*

  This option is by default set to Target Directory. This option determines where the generated files will be put in the file system.

- *File name prefix*

  The names of the generated files will be prefixed by the text put in this field.

Dialog Buttons

> **Note:**
>
> For information about how *Make/Full Make* relates to *Analyze/Full Analyze* and quick buttons, see "Normal versus full analyze (make)" on page 113.

- *Make*

  Starts the code generation in the background and closes the dialog. Status from the ongoing Make process is shown in the Organizer Log window. First the time stamps of the SDL files, e.g *.sbk, *.spr, *.spd, etc, are compared with all their dependent generated c-file counterparts. Only those diagrams that have a newer timestamp are converted from SDL GR to PR, analyzed, and code is generated. After any needed code is generated the make function corresponding to the specified compiler is called to compile and/or link the system.

Situations where a make will prove useful are when the following circumstances exist:

- Diagrams had errors the last time the system was analyzed.
- Diagrams were changed.
- Diagrams for which the target files have been removed.
- Diagrams that any diagram mentioned above are depending on.

- *Full Make*

  Has the same effect as the *Make* button, except that it forces the entire system to be regenerated, even if only certain parts needs to be regenerated. After any needed code is generated the make function corresponding to the specified compiler is called to compile and/or link the system.

- *Set*

  Saves the Make option settings and closes the dialog, but does not perform a make.

- *Analyze Options*

  Opens the *Analyze* dialog to set the analyze options. The *Set* button returns to the *Make* dialog.

TTCN Make

This menu choice is used to make (generate code, compile and link) a TTCN system.

For more information about this operation, see chapter 28, *The TTCN to C Compiler (on UNIX)* or chapter 33, *The TTCN to C Compiler (in Windows)*.

Stop Analyze/Make (UNIX only)

This menu choice stops an ongoing analyze/make operation. The Analyzer tool is also stopped. (The Analyzer tool normally remains resident in memory for the rest of the SDL suite session once the first analyze/make is performed, and this menu choice is then renamed to *Stop Analyzer*. Using this menu choice is one way to free memory if needed.) Several commands, such as *Analyze*, *Make*, *Convert to PR/MP* and

Convert to GR, are not available when the Analyzer is processing data. Stopping the Analyzer enables these commands again.

A message with the essence “Analyzer could not be stopped” may be issued as a response to this command; in this case, repeat the menu choice until the message “Analyze/make stopped” is issued in the message area.

Targeting Expert

This menu choice starts the Targeting Expert tool. See chapter 60, *The Targeting Expert* for more information.

SDL Overview

This menu choice generates an SDL overview diagram for the selected SDL diagram as the top diagram. The menu choice is dimmed if:

- No SDL diagram is present in the Organizer.
- The selected SDL diagram is not connected.
- The selected icon is marked invalid.
- The system file is read-only.

If modified information exists in the current system structure, the user should first save it; see “The Save Before Dialog” on page 64.

Any SDL diagram can be selected for generation. If no SDL diagram is selected, the first SDL diagram found in the Organizer view will be used.

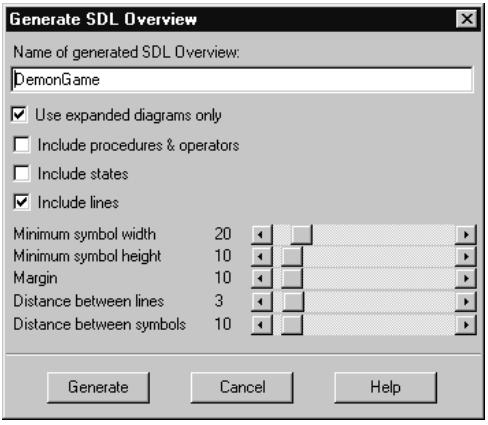

Figure 39: The Generate SDL Overview dialog

- *Name of Generated SDL Overview*

  Specifies the name of the generated Overview diagram. The Organizer proposes a name based on the SDL diagram which the Overview should be generated from.

- *Use Expanded Diagrams Only*

  Controls whether the resulting SDL Overview should comprise all diagrams that are included in the SDL diagram substructure, or only include the diagrams which are expanded (i.e. visible in the Organizer).

- *Include Procedures & Operators*

  Governs if SDL procedure and operator diagrams should be included in the SDL Overview.

- *Include States*

  Controls whether state symbols should be added to the SDL Overview or not.

- *Include Lines*

  Governs whether lines (such as channels and signal routes) should be added to the SDL Overview or not.

- *Minimum Symbol Width*

- *Minimum Symbol Height*

  These slide bars control the minimum size the tool will apply on symbols. (Symbols will be shrunk, if possible, to make the SDL Overview smaller.)

- *Margin*

  This slide bar specifies the distance (in any direction) between symbols at one SDL level and the enclosing frame (the boundaries of the enclosing SDL symbol).

- *Distance between Lines*
- *Distance between Symbols*

  These slide bars define what distance will be inserted between lines that otherwise will overlap each other in the generated SDL Overview diagram.

- *Generate*

  Causes the generation of the overview diagram to start. The user is informed about the progress of the generation in the Organizer Log window. The generated SDL Overview is added as a root document directly after the SDL structure for which the Overview diagram is generated. An SDL Editor opens and presents the Overview diagram.

State Overview

This menu choice is used to generate a state overview information file from an SDL system or a group of state charts. The state overview can be viewed either as state matrices in the Text Editor or as state charts in the State Chart Editor.

Before selecting this menu choice, make sure that the information source (an SDL system or one or more state charts) is selected directly or indirectly. For instance, you can select a group of state charts by selecting the chapter symbol that contains the state charts.

When you select this menu choice, a dialog is displayed where you can:

- Decide if the information source should be SDL or state charts.
- Change SDL system, if several SDL systems were selected in the Organizer. Use the first *change* button for this.

- Pick out individual state charts, among the state charts that were selected in the Organizer. Use the second *change* button for this.
- Decide name and directory for the generated state overview information file.
- Decide if an Organizer symbol should be created.
- Decide if the Text Editor should show the state overview information as state matrices.
- Decide if the State Chart Editor should show the state overview information as state charts.

CPP2SDL Options

This menu choice is available if a C or C++ Import Specification is selected. For more information, see "The CPP2SDL Tool" on page 757 in chapter 15, *The CPP2SDL Tool*.

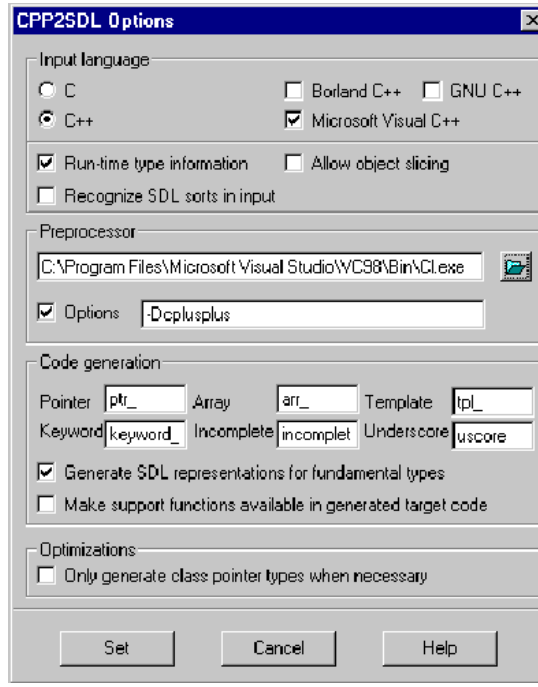

Figure 40

Convert to PR/MP

This menu choice converts the selected diagram/document to textual form. SDL/GR is converted to SDL/PR and TTCN-GR is converted to TTCN-MP. If no diagram/document is selected, the Organizer operates on the first SDL system found in the Organizer. The menu choice is dimmed if a job using the Analyzer is already running.

The *Convert to PR (SDL)* variant of the menu choice is described below, followed by the *Convert to MP (TTCN)* variant.

Convert to PR (SDL)

The SDL variant of the *Convert to PR/MP* menu choice generates a formatted (pretty printed) SDL/PR file. Input is either an SDL/PR file or an SDL/GR diagram structure.

If modified information exists in the current system structure, the user should first save it; see "The Save Before Dialog" on page 64.

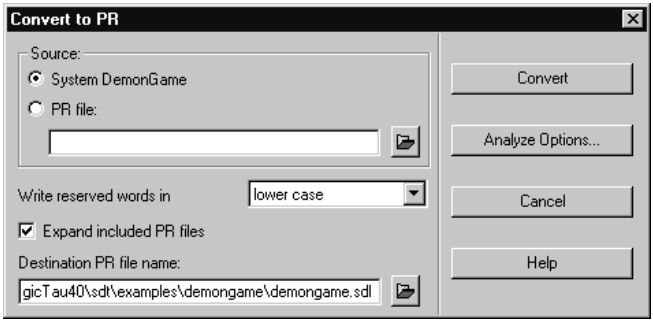

Figure 41: The Convert to PR dialog

- *Source:*
  <type> <name>

  Converts an SDL/GR diagram structure to PR. The top of the diagram structure appears in the name of the button.

  If an SDL diagram is selected, that diagram will be converted. Otherwise, the first SDL diagram structure found in the Organizer will be converted. If no SDL diagrams are found, this option is dimmed, in which case the name of the button is "<No SDL/GR diagram selected>". This will also be the case if the SDL diagram is not connected, or if its icon is marked invalid.

  If the SDL diagram to be converted has an expanded diagram substructure visible in the Organizer, this substructure is also converted. All PR code is put in the destination PR file.

- *PR file*

  Generates a pretty printed PR file from an existing PR file, which is specified in the input field. By default, this file is read from the Source Directory.

- *Write reserved words in*

  Specifies whether reserved words in SDL are to be written in lower case or upper case.

- *Expand included PR files*

  Expands SDL/PR include files found in `/*#INCLUDE...*/` comments.

- *Destination PR file name*

  Specifies the pretty printed PR file to generate. The default name uses the selected diagram name as prefix and `.sdl` as extension. By default, this file is stored in the Target Directory.

  If no file name is provided, the user is warned and no conversion is performed.

  An Overwrite confirmation dialog is issued if the user changes the suggested file name and specifies a file that already exists.

- *Convert*

  Generates the pretty printed PR file and closes the dialog. Status from the ongoing conversion process is shown in the Organizer Log window.

- *Analyze Options*

  Opens the *Analyze* dialog to set the analyze options. The *Set* button returns to the *Convert to PR/MP* dialog.

Convert to MP (TTCN)

This TTCN variant of the *Convert to PR/MP* menu choice converts a TTCN-GR document to a TTCN-MP text file.

For more information about *Convert to MP*, see "Exporting a TTCN Document to TTCN-MP" on page 1144 in chapter 25, *The TTCN Browser (on UNIX)* or "Converting to TTCN-MP" on page 1266 in chapter 31, *Editing TTCN Documents (in Windows)*

Convert to GR

This menu choice converts a textual file to one or more graphical diagrams. A TTCN-MP file is converted to TTCN-GR diagrams, and an SDL/PR file is converted to SDL/GR diagrams. The menu choice is dimmed if a job using the Analyzer is already running.

The *Convert to GR (SDL)* variant is described below, followed by the *Convert to GR (TTCN)* variant.

Convert to GR (SDL)

The *Convert to GR* dialog is in SDL mode when the radio button *Convert* SDL/PR to SDL/GR is on (see Figure 42).

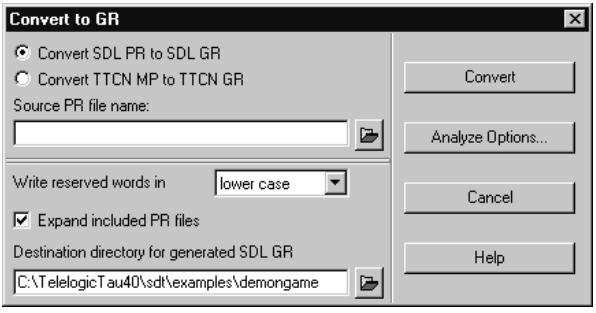

Figure 42: The Convert to GR dialog in SDL mode

- *Source PR file name*

  Specifies the PR file to convert.

- *Write reserved words in*

  Specifies whether reserved words in SDL are to be written in lower case or upper case.

- *Expand included PR files*

  Expands SDL/PR include files found in `/*#INCLUDE...*/` comments.

- *Destination directory for generated SDL/GR*

  Specifies the directory where to put the generated SDL diagrams. By default, the Source Directory will be used (see "Set Directories" on page 70).

- *Convert*

  Generates the SDL/GR diagrams. The SDL/GR files are named in the same way as when saving unconnected documents in the Orga-

nizer, i.e. files are named so that no files are overwritten. Status from the ongoing conversion process is shown in the Organizer Log window.

- *Analyze Options*

  Opens the *Analyze* dialog to set the analyze options. The *Set* button returns to the *Convert to GR* dialog.

Convert to GR (TTCN)

The *Convert to GR* dialog is in TTCN mode when the radio button *Convert TTCN-MP to TTCN-GR* is on (see Figure 42).

For more information about the Convert to GR dialog in TTCN mode, see "Importing a TTCN-MP Document" on page 1149 in chapter 25, *The TTCN Browser (on UNIX)* or "Converting to TTCN-MP" on page 1266 in chapter 31, *Editing TTCN Documents (in Windows)*.

Convert GR to CIF

This menu choice converts SDL/GR diagrams to the Common Interchange Format (CIF).

On UNIX, the *Convert GR to CIF* dialog is opened. For more information, see "Convert GR to CIF Dialog (UNIX only)" on page 898 in chapter 17, *CIF Converter Tools*.

In Windows, the SDT2CIF converter tool is started. For more information, see "Graphical User Interface (Windows only)" on page 900 in chapter 17, *CIF Converter Tools*.

Convert CIF to GR

This menu choice converts diagrams in Common Interchange Format (CIF) to SDL/GR diagrams.

On UNIX, the *Convert CIF to GR* dialog is opened. For more information, see "Convert CIF to GR Dialog (UNIX only)" on page 883 in chapter 17, *CIF Converter Tools*.

In Windows, the CIF2SDT converter tool is started. For more information, see "Graphical User Interface (Windows only)" on page 884 in chapter 17, *CIF Converter Tools*.

Convert State Chart to SDL

This menu choice transforms the selected State Chart to an SDL process diagram. For more information, see "Converting State Charts to SDL" on page 1658 in chapter 40, *Using Diagram Editors*.

Edit Separation

This menu choice inserts or edits a separation on the selected SDL diagram. It is dimmed if the selected diagram type is not one of system, system type, block, block type, process, process type, procedure, and package. This menu choice is only available if the system file can be changed.

Diagram separation symbols are used during code generation and controls both the splitting of the target into separate modules and the naming of these modules. For more information, see "Selecting File Structure for Generated Code – Directive #SEPARATE" on page 2650 in chapter 57, *The Cadvanced/Cbasic SDL to C Compiler*. Separations can be shown in the diagram structure of the Organizer; see "Separator symbols" on page 108. Information about separations are stored in the system file.

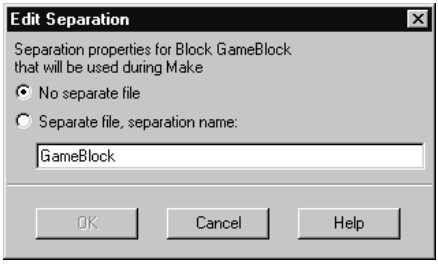

Figure 43: The Edit Separation dialog

- *No separate file*

  No separation is to be used on this diagram; any existing separation is cleared. This option is disabled on system and package diagrams.

- *Separate file*

  Inserts a separation on this diagram, i.e. the diagram and its substructure is generated separately. In the text field, the name of the

separation is specified, i.e. the prefix of the files generated for this separation.

Dependencies

This menu choice introduces or removes dependencies between a selected document and other documents. A dependency symbol below a document indicates that the document is depending on another document. For instance, if an SDL system is depending on an ASN.1 document, then the SDL system must be re-analyzed each time the ASN.1 document is updated. A related symbol is the association symbol, see "Associate" on page 97.

This menu choice is only available if the system file can be changed.

Note:

Dependency links between SDL systems and C header files or ASN.1 documents are **not** required, but serve mainly as comments. The SDL Analyzer will re-analyze the SDL system automatically for such dependencies.

If a dependency icon is selected, this menu choice operates on the document the dependency icon is referring to, not the icon itself. Any document may depend on any other document, and a document may have be depending on more than one document.

The following dialog is opened:

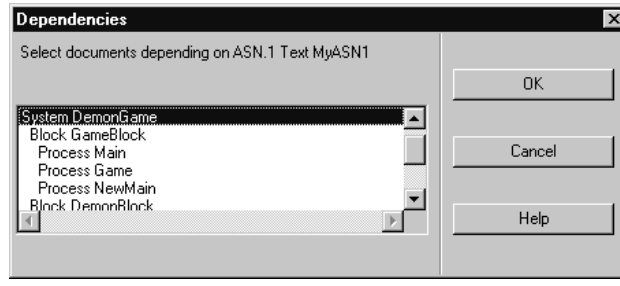

Figure 44: The Dependencies dialog

- *Select documents depending on <document>*

The multiple selection list displays the type and name of all documents in the Organizer structure. When the dialog is opened, all documents that the current document is depending on are selected, i.e. the list shows all dependencies for the document.

By selecting a new document in the list, a dependency for the current document will be created. By deselecting a document in the list, the corresponding dependency link will be removed from the selected document's structure.

Merge ASN.1

This menu choice controls whether the ASN.1 files in an Organizer module should be merged (joined) into a single SDL package or not. For more information, see "Using the ASN.1 Utilities" on page 697 in chapter 14, *The ASN.1 Utilities*.

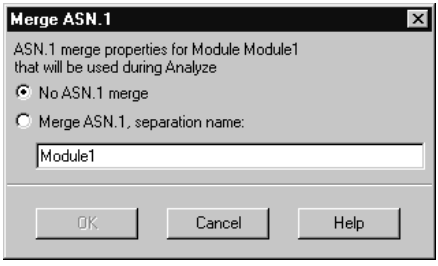

Figure 45: The Merge ASN.1 dialog

- *No ASN.1 merge*

  The ASN.1 files in this Organizer module should not be merged; a separate SDL package is generated for each of the ASN.1 files.

- *Merge ASN.1, separate name*

  The ASN.1 files in this Organizer module should be merged; one SDL package will be generated, containing all ASN.1 modules from the ASN.1 files. The name of the SDL package will be the same as the name of the Organizer module, and the separate name will be used for the code generation in a similar way as for ordinary ASN.1 files.

Tools Menu

The *Tools* menu contains the following menu choices:

- *Organizer Log*
- *Link > Create Endpoint*
- *Link > Traverse*
- *Link > Link Manager*
- *Link > Clear Endpoint*
- *Search*
- *Spelling > Comments*
- *Spelling > All Text*
- *Change Bars*
- *Compare SDL Diagrams*
- *Merge SDL Diagrams*
- *Split*
- *Join*
- *Compare State Machines*
- *Simulator Test > New Simulator*
- *Simulator Test > Existing Simulator*
- *Editors > Deployment Editor*
- *Editors > HMSC Editor*
- *Editors > MSC Editor*
- *Editors > OM Editor*
- *Editors > SDL Editor*
- *Editors > State Chart Editor*
- *Editors > Text Editor*
- *Editors > TTCN Browser*
- *SDL > Type Viewer*
- *SDL > Coverage Viewer*
- *SDL > Index Viewer*
- *SDL > Simulator UI*
- *SDL > Validator UI*
- *SDL > Target Tester UI*
- *TTCN > Find Table*
- *TTCN > Access*
- *TTCN > Simulator UI*
- *UML Suite*
- *Preference Manager*

Organizer Log

This menu choice raises the Organizer Log window. The Organizer Log window can be raised automatically when the user performs an analysis or other forms of generation. There is only one Organizer Log window.

The Organizer Log window is described in "Organizer Log Window" on page 179.

Link > Create Endpoint

This menu choice creates an endpoint for the selected document. A document with an endpoint is recognized by a small triangle in the upper left corner of the document symbol. See "Link > Create Endpoint" on page 444 in chapter 10, *Implinks and Endpoints*.

This menu choice is only available if the system file can be changed.

Link > Traverse

This menu choice traverses a link for the selected document. A document with at least one link is recognized by a small black triangle in the upper left corner of the document symbol. See "Link > Traverse" on page 444 in chapter 10, *Implinks and Endpoints*.

Link > Link Manager

This menu choice opens the Link Manager's main window. For more information about the Link Manager, see "The Link Manager" on page 462 in chapter 10, *Implinks and Endpoints*.

Link > Clear Endpoint

This menu choice removes an existing endpoint for the selected document. See "Link > Clear Endpoint" on page 447 in chapter 10, *Implinks and Endpoints*.

This menu choice is only available if the system file can be changed.

Search

This menu choice searches for text in SDL, MSC, HMSC, Object Model, and State Chart diagrams. Textual documents (C Header, Text ASN.1, and Text Plain) are also searched if the Text Editor is used. (The Text Editor is used if the preference variable SDT*TextEditor is set to

"SDT.") It is not possible to search in TTCN, Word or Generic documents with this menu choice. To search in TTCN documents, use *TTCN > Find Table*.

The document scope of the search depends on the selection:

- If a module is selected, all searchable documents in that module will be searched.
- If a chapter is selected, all searchable documents in that chapter will be searched.
- If a root document is selected, all searchable documents in that document's substructure will be searched.
- If there is no selection, all searchable documents in the Organizer will be searched.

The menu choice is dimmed if the selected document is not a root document, not connected, or marked invalid.

The search will only take place in diagrams that are connected and do not have an invalid status. The search will start in the selected diagram and continue in top-down order for the rest of the diagrams (the order is left-right in a tree view).

The search function will go through the list of diagrams and stop each time the search criteria, as set in the dialog below, matches. If a search/replace string or any option is changed when the search is stopped (a match is found or the user pressed *Abort*), these values become the basis when the search is continued.

The search process will open an editor window, if necessary, and select the matched search text.

The searching is based on ASCII character matching. All text fragments in symbols are searched, with a few restrictions (see below).

When all diagrams have been searched, a beep is issued and the message *Search completed* appears.

Dialog Fields and Options

The *Search* dialog contains the following fields and options.

- *Search for*

The text to search for. To the left of the text field, there is an option menu containing old search strings. To search again for a string that has already been searched for, select the search text in the option menu.

- *Replace with*

  The text which is to replace the text searched for. Does not have to be specified.

- *Search in*

  Two option menus where it is possible to restrict the search to a selected diagram or symbol type. For instance, you can find all SDL input symbols by not specifying any search text, and specifying "SDL" as a diagram type and "Input" as a symbol type.

- *Consider case*

  If this option is set, search is case sensitive.

- *Wildcard search*

  Specifies whether a wildcard matching will be used. In wildcard search the asterisk ('*') matches a sequence of zero or more characters of any kind, e.g. whether the search text *dist*ution* will find the text *distribution* or not.

- *Search substructure*

  If this option is set, not only the selected diagram but also its diagram substructure will be searched.

Note:

Only the textual elements that are visible in an MSC will be searched. See "Diagram Options" on page 1632.

Dialog Buttons

When the dialog is first opened, all buttons except *Replace&Search* and *Replace All* are enabled. When *Search*, *Replace&Search* or *Replace All* is pressed all fields and buttons are disabled except the *Close* button. (The *Close* button changes name during the search to *Abort*.) If a search string is found in an editor, it is selected and all buttons and fields are enabled.

When the first search or replace operation has been applied and control returns to the *Search* dialog, it is possible to perform a new search on the same diagram(s).

- *Search*

  Searches for the search string. An editor is opened when the search string is found. If no text is supplied in the *Search for* field, a warning message appears. Confirming the message box will return control to the *Search* dialog.

- *Replace&Search*

  Replaces the current match with the replace string and searches for the next match. An editor is opened when the search string is found.

- *Replace All*

  Replaces all occurrences of the search string with the replace string. No editors are opened. However, diagrams with replaced text are loaded in the editor and gets a "dirty" state in the Organizer.

  Text in reference symbols and in kernel headings are not replaced.

- *Close/Abort*

  The *Close* button closes the *Search* dialog. The *Close* button is temporarily renamed to *Abort* during search in diagrams. If *Abort* is pressed, the current search is stopped as soon as possible (when a new file is to be searched). After an *Abort*, the dialog remains on screen, ready for new input.

Search Restrictions

All data in the editors that affects the diagram structure maintained by the Organizer (primarily reference symbols and kernel headings in the SDL Editor) is regarded as **read only** during the search operation. That is, they are not affected by the search.

Externally editing (i.e. through means other than using replace) of a diagram during a search operation completely resets the search, i.e. the next search starts from the first diagram.

The search may fail if dialogs are opened in the editor during the search. In this case the editor blocks the continuation of the search process. To continue the search process, the editor dialog must be closed.

The Organizer's data is locked during the search process. This is normally not noticed since the *Search* dialog is modal, but the SDL Editor needs to access that data to perform operations affecting the diagram structure. The Organizer will deny the editor's requests to modify the data structure. The duration of the search process is the period of time during which the *Search* dialog is visible in the Organizer.

Fast Search

Fast search is invoked by pressing ctrl+F on the keyboard.

Fast search behaves like normal search, except that:

- No dialog is used.
- The replace functionality can not be used.
- Consider case is always off.
- Wildcard search is always off.
- Search substructure is always on.

When Fast search is invoked, the message area displays the text that will be searched for. Initially, the text is *Search for:*. When you type on the keyboard with the mouse pointer over the drawing area, the characters will turn up in the message area. When you have typed the text pattern to search for, press enter or return to start the search operation.

The same search operation as for normal Search is used. If the search operation finds a match, you can search for another match by pressing enter or return once more in the Organizer window.

To finish the Fast search operation, click in the drawing area or select another operation.

The next time Fast search is invoked with ctrl+F, the search text that was used the last time is proposed as a search text once more. To use it, press enter or return. To use another search text, press ctrl+F once more or the delete key several times, to erase the search text.

Spelling > Comments

Check the spelling of comments in selected diagrams. Comments can be either comment symbols or /* C-style comments */

Note:

This command only works if the spelling checker *ispell* has been installed on your computer, and the preference SDT*ISpellCommand correctly identifies the *ispell* executable. *ispell* can be found on the internet and uses the same kind of license as the *emacs* text editor.

For spelling errors, the Spelling dialog appears, with the following possibilities:

- *Word replacement text field*. Initially, the word from the diagram is presented here. One way to correct the spelling mistake is to correct the spelling in this text field and press the *change* button.
- *List of suggestions*. For most words, the spelling checker will present possible corrections here. Click on a word in this list to update the *word replacement text field*. Double-click on a word to update the diagram.
- *Add button*. Add the word currently in the *word replacement text field* to your personal dictionary.
- *Change button*. Update the diagram with the word from the *word replacement text field* and find the next spelling error.
- *Ignore button*. Ignore the current spelling error and find the next spelling error.
- *Ignore All button*. Ignore all occurrences of the current spelling error during this session and find the next spelling error. For more "permanent ignorance", use the *Add button*.

Spelling > All Text

Works in the same way as *Spelling > Comments*, but all text is checked instead of just comment texts.

Change Bars

This menu choice is used to control the usage of change bars in SDL diagrams. A dialog with two options is opened:

- *Reset change bars for selected SDL diagrams*

  This option is used to reset change bars in SDL diagrams that fall under the selection in the Organizer. Note that removing a change

bar makes the diagram dirty, i.e. change bars cannot be removed from diagram files where you do not have write access. If this option is on, a second dialog will be opened telling you about the diagrams that the operation will be applied upon.

- *Create change bars when SDL changes occur*

  This option is used to decide if change bars should be created when an SDL diagram is edited. This option is saved in the system file and is valid for all SDL diagrams in the system. The preference *ChangeBars* is used as default for new SDL systems. See "ChangeBars" on page 243 in chapter 3, *The Preference Manager*.

Compare SDL Diagrams

This menu choice compares the contents of SDL diagram file pairs. A diagram file pair is constructed by matching an SDL diagram file loaded into the Organizer with an SDL diagram file with the same name, but in a different directory.

> **Note:**
>
> There is a similar operation in the SDL Editor for comparing one SDL diagram pair at a time. See "Compare Diagrams" on page 1961 in chapter 44, *Using the SDL Editor*.

The menu choice is dimmed if there are no SDL diagrams in the Organizer.

In the same way as for the *Search* menu choice, the compare operation is limited by the selection in the Organizer. Only SDL diagrams within the scope of the selection will be considered for the compare operation.

When the Compare SDL Diagrams menu choice is invoked, the compare SDL diagrams setup dialog appears.

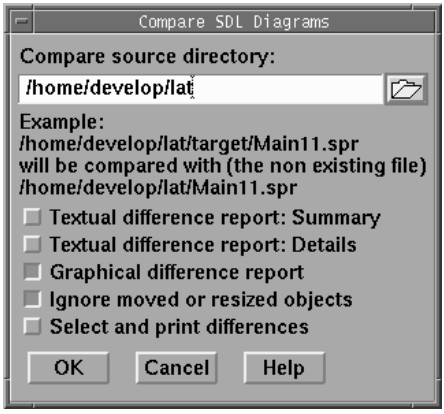

Figure 46: The Compare SDL Diagrams Setup dialog

In the compare SDL diagrams setup dialog, the following input parameters to the compare operation can be specified:

- *Compare source directory*

  The directory specified here is used instead of the normal source directory to find SDL diagrams to compare the SDL diagrams in the Organizer with.

  – For SDL diagrams in the Organizer that are relative to the normal source directory, the relative part of the file path is kept, but the part of the file path that matches the normal source directory is replaced with a part of a path constructed from the diff source directory instead.

    For example, if the source directory is `/home/lat` (`E:\home\lat` **in Windows**) and the diff source directory is `/opt/home/gn` (`E:\opt\home\gn`), then the Organizer file `/home/lat/demo/x.ssy` (`E:\home\lat\demo\x.ssy`) will be compared with `/opt/home/gn/demo/x.ssy` (`E:\opt\home\gn\demo\x.ssy`), if that file exists.

  – For SDL diagrams in the Organizer that are not relative to the normal source directory, the directory part of the file path is replaced with diff source directory.

For example, if the source directory is `/home/lat` (`E:\home\lat` **in Windows**) and the diff source directory is `/opt/home/gn` (`E:\opt\home\gn`), then the Organizer file `/usr/local/lat/y.ssy` (`E:\usr\local\lat\y.ssy`) will be compared with `/opt/home/gn/y.ssy` (`E:\opt\home\gn\y.ssy`), if that file exists.

- *Textual difference report: Summary*

  When this option is chosen, a summary of found differences is printed for each diagram pair that is compared. If more than one diagram pair is compared, a summary for all compared diagram pairs is also printed. The textual difference report is printed in the Organizer Log window.

- *Textual difference report: Details*

  When this option is chosen, detailed information about every found difference is printed as a textual report in the Organizer Log window.

- *Graphical difference report*

  When this option is chosen, differences are shown in the SDL Editor, one difference group at a time.

- *Ignore moved or resized objects*

  When this option is chosen, the compare operation tries to ignore reporting differences only caused by moved or resized symbols by using symbol ids. This option should only be used when comparing two versions of the same original diagram file: moved symbols cannot be accurately detected by using symbol ids when comparing two versions of a diagram that are both built from scratch.

- *Select and print differences*

  When this option is chosen, pages from the compared diagram versions are printed, but only those pages that contain differences.

When the OK button in the dialog is pressed, the Organizer checks if a matching diagram can be found for all SDL diagrams in the operation. If that is not the case, a dialog appears to inform about this fact.

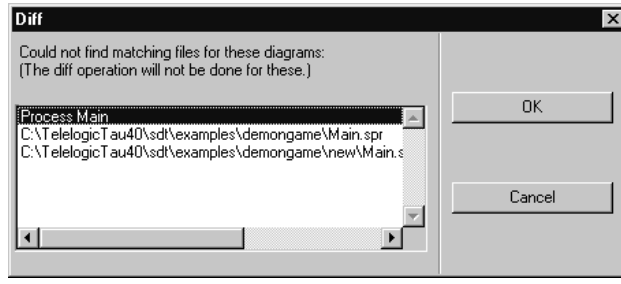

Figure 47: The Compare SDL Diagram missing files dialog

Finally, the real compare operation starts in the SDL Editor. Read more about this in "Compare Diagrams" on page 1961 in chapter 44, *Using the SDL Editor*.

Merge SDL Diagrams

This menu choice compares the contents of SDL diagram file pairs in the same way as "Compare SDL Diagrams" on page 146 does. The main difference is that this menu choice gives the possibility to merge differences: For each diagram pair, a new merge result diagram is created. Read more about how the actual merge is performed in "Merge Diagrams" on page 1961 in chapter 44, *Using the SDL Editor*.

Split

This menu choice is used to split one SDL diagram into two SDL diagrams. This menu choice can be applied several times to split one SDL diagram into several parts. This menu choice is, together with the menu choice *Join*, useful in situations where several people have to work simultaneously on one SDL diagram. The SDL diagram file is partitioned into several SDL diagram files and each person is given one part to work on.

To split an SDL diagram consisting of several pages, select the diagram symbol in the Organizer and select the *Split* menu choice. (It is also possible to select a page symbol associated with the diagram that should be split.)

A dialog appears with a list of all pages but the first one in the diagram, see Figure 48. Select a page to define how the SDL diagram should be split. All pages before the selected page will end up in the first SDL di-

agram part. The selected page and all pages after the selected page will end up in the second SDL diagram part.

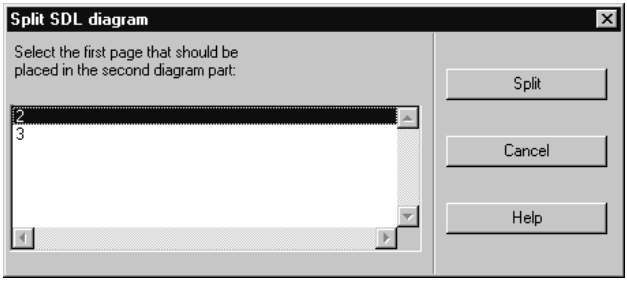

Figure 48: The first Split dialog

The Split button in the first Split dialog closes the dialog and brings up the second Split dialog, as shown in Figure 49. The second Split dialog is used to specify the files that the two resulting SDL diagram parts should be saved in.

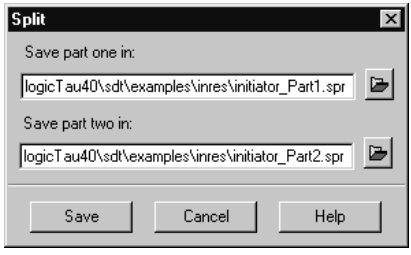

Figure 49: The second Split dialog

When the second Split dialog is closed with the *Save* button, the split operation is performed: The SDL Editor is loaded with the SDL diagram that should be split. Two new smaller SDL diagrams are created in the SDL Editor and saved under the file names specified in the second Split dialog.

The visible result of the Split operation is that the SDL Editor contains at least the diagram that was split and the resulting SDL diagram parts. Note that no additional symbols are created in the Organizer. The SDL diagram parts are accessed by opening the SDL diagram part files in the SDL Editor with the *Open* menu choice. It is of course possible to man-

ually add symbols for the SDL diagram parts in the Organizer with for instance the *Add Existing* menu choice.

Join

This menu choice is used to join two SDL diagrams of the same type into one SDL diagram. This menu choice is, together with the *Split* menu choice, useful when several people have to work on the same SDL diagram at the same time.

Note that it is not necessary to do a Split before doing a Join. One way to work in parallel on the same diagram is to let one designer work on the existing SDL diagram, while another designer creates new pages destined for the same SDL diagram, in a new SDL diagram with the same type as the existing SDL diagram. A join operation when the parallel work is finished puts the new SDL pages in the correct SDL diagram.

The resulting SDL diagram is produced by copying the complete first diagram part and merging/joining pages from the second diagram part by copy and paste. This means that information that is common for all pages in an SDL diagram is taken from the first diagram part. This includes:

- The diagram name and any other information in the header.
- The information in the extended heading symbol.
- The information in the use text symbol.

Another consequence of this way of working is that duplicate page names and reference symbol names emerging from the second diagram part are changed by the join operation to make them unique. Auto-numbered pages from the second diagram part will also have their names changed.

When the *Join* menu choice is invoked, the Join dialog appears, see Figure 50.

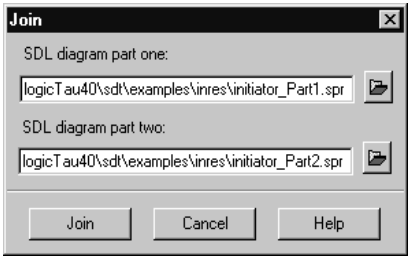

Figure 50: The Join dialog

In the dialog, specify the names of the two SDL diagram files that should be merged. Pressing the *Join* button will close the dialog and start the Join operation.

The result of the Join operation is that three diagrams will be loaded in the SDL Editor; the two SDL diagram parts that act as input to the Join operation and a new and unsaved SDL diagram that contains all the pages from both the input diagrams. The first thing that you will normally do after a Join operation is to save the new SDL diagram in a file. After that, it might be appropriate to check the new SDL diagram in the context of the SDL system with the Analyzer.

Compare State Machines

This menu choice is used to compare state machines with other state machines. This menu choice makes it possible to keep state machines expressed as a group of state charts consistent with state machines expressed as processes in an SDL system.

Before selecting this menu choice, make sure that the information sources (SDL systems and/or state charts) are selected directly or indirectly. For instance, to compare the only SDL system in the Organizer with all state charts in the Organizer, make sure that there is no selection in the Organizer by clicking in the background. The Organizer will interpret this as “everything is selected”.

When you select this menu choice, a dialog is displayed where you can:

- Specify the two groups of state machines that should be compared. A group of state machines is either the state machines in an SDL system or a group of state charts. It is only possible to choose among

the state machines that were selected in the Organizer. Use the *Select* buttons to select individual SDL systems or state charts.

- Specify the two state overview information files that will be generated and compared.
- Specify if SDL procedure calls and/or state chart call actions should be compared.
- Specify if SDL outsignals and/or state chart send events should be compared.

When the dialog is closed, two state overview information files are generated. A state overview describes a state machine in a normalized form:

- For an SDL system, diagram type inheritance and diagram type instantiation have been removed.
- For a state chart, state hierarchies (states in states) have been removed. The same rules as when converting a state chart to SDL are used, for more information, see "Converting State Charts to SDL" on page 1658 in chapter 40, *Using Diagram Editors*.

The compare operation compares the two generated state overview information files in the following way:

- State machines are matched by their names. For instance, SDL process A is matching state chart A.
- States are matched by their names.
- Transitions are matched by a combination of from-state name, in-signal name and to-state name

Note:

The compare operation does not compare all the details for a transition. For instance, the conditional expression in an SDL decision symbol is not compared with the guard condition in a state chart transition.

If the compare operation finds anything that does not match, this is reported in the Organizer Log. You use the Organizer Log quick button *Show Error* to navigate to the SDL diagram or state chart with an entity that did not match anything in the other group of state machines.

Simulator Test > New Simulator

This menu choice is used to execute test cases in the simulator for an SDL system. Test cases can be described either as MSCs or as simulator UI input scripts (*.cui). A description of how to express MSCs in this context can be found in *Using MSCs as test cases*.

When this menu choice is invoked, a dialog appears with a list of all MSCs and input scripts. In the dialog, it is possible to select test cases that should be executed. The selection in the Organizer decides:

- primarily the SDL system to be used,
- but also the default test case selection in the dialog.

When the dialog is closed, an SDL simulator is generated and each selected test case is run. When all test cases have been executed, the Organizer Log will contain a one-line summary for each test case, with information about if the test case passed or failed. For test case failures, the MSC editor will pop up, showing the symbol in the MSC that failed (this happens only if the test case was expressed as an MSC). To be able to examine test case failures in detail, run each test case that fails separately, because then the MSC editor will show the place of the MSC test case failure and the textual output from the simulator will contain information about the test case failure.

Using MSCs as test cases

MSC test cases are MSCs written in a special way. MSC test cases are high-level test cases that are auto-converted to low-level simulator UI input script test cases before they are executed.

When MSCs are used as test cases, you can:

- send in a signal to the system and check that the expected signal is received as a response from the system.
- check that a process instance is in a certain state.
- check that a process instance variable has a certain value.
- check if a process instance exists.
- ...

Figure 51 illustrates MSC test case building blocks that are described in the text below.

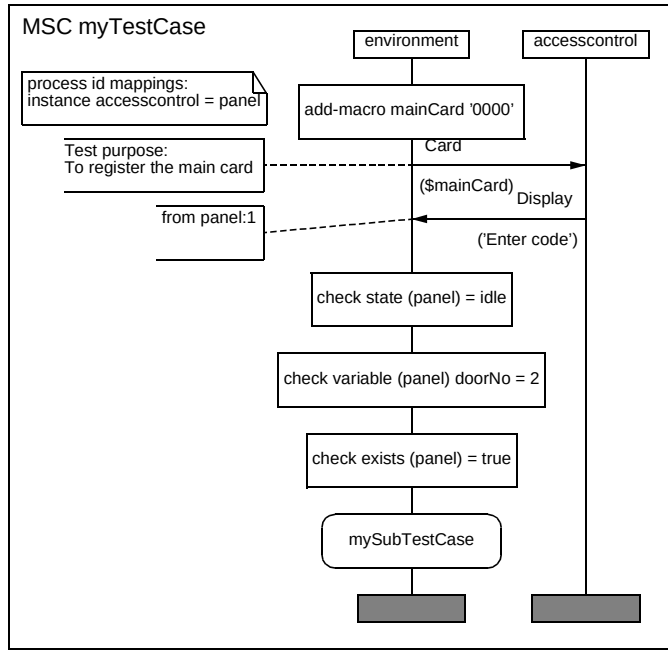

Figure 51 MSC test case building blocks

MSC instance symbols are used in the following way in MSC test cases:

- Use an MSC instance symbol called *environment* or *env_1* or something similar as an instance to send and receive signals to and from the SDL system.

  By using an .itt file other instances than environments can be mapped to act as an environment during testing. See "Mapping Instances to Different Environments" on page 2160 in chapter 50, *The SDL Simulator*.

- Use one or several MSC instance symbols to represent the SDL system. If you use one instance, give it the same name as the SDL system (for instance *accesscontrol*). If you use several instances, give each the name of an SDL process instance that you want to send and receive signals to and from (for instance *panel* or *panel:1*).

To send a signal into the system, draw an MSC signal from the environment instance to one of the instances representing the SDL system. Specify the signal name and any parameters that you know about.

To check that a signal is sent out from the system, draw an MSC signal from an instance representing the SDL system to the environment instance. Also here, you should specify the signal name and any parameters. The check is done by matching a text string created from the MSC with the actual textual output from the simulator. You can use the * character (matching any characters) if you for instance do not want to specify all parameters in the MSC.

There are several overlapping ways of specifying the process instance a signal should be sent to or from. Two of them involves a mapping table in an MSC text symbol. The text should look something like this (one line with a '=' for each mapping rule):

```
process id mappings:
instance accesscontrol = panel
signal display = panel
```

- First of all, the SDL suite finds out if there is a comment symbol connected to the signal. If there is, and the text is something like *to panel* or *from panel*, the SDL suite uses panel as the process instance name (with *:1* added because no *:* with a number was specified).
- Second, the SDL suite finds out if there is a signal mapping rule matching the current signal name. If there is, the process instance name in that rule is taken.
- Third, the SDL suite finds out if there is an instance mapping rule matching the instance name of the instance representing the SDL system that is attached to the current signal. If there is, the process instance name in that rule is taken.
- If no process instance name has been found yet, then the MSC instance name of the instance representing the SDL system is taken as the SDL process instance name.

It is possible to include any simulator command in an MSC test script by attaching an MSC action symbol to the environment instance, and typing the simulator commands in it. One line for each command.

This can for instance be used to declare macros representing parameter values. If you have done `add-macro myMacroName 5` in an action sym-

bol, you can type `$myMacroName` instead of `5` as the value of a parameter. Macros can be defined in a separate MSC and used in normal MSC test cases. Just make sure that the macro MSC is executed before the normal MSC test cases when you do a simulator test.

There are three textual shortcuts that can be used in an action symbol:

- To check that an SDL process instance is in a certain state:

```
check state (panel) = IDLE
```

- To check the value of a variable in an SDL process instance:

```
check variable (panel) DoorNo = 1
```

- To check if an SDL process instance exists or not:

```
check exists (panel) = false
```

The MSC reference symbol can be attached to the environment instance. Type in a name of a sub MSC test script that should be executed. This is a way to avoid repeating the same information in many places. It can for instance be used in the beginning of a test script to perform common initialization of the system.

Simulator Test > Existing Simulator

Same as "Simulator Test > New Simulator" on page 154, except that an already created simulator is used. Before the normal simulator test dialog (used to specify test cases to execute), a file selection dialog appears, where a simulator executable can be specified.

Editors > Deployment Editor

Adds a Deployment diagram symbol to the Organizer view, and starts the Deployment Editor. The symbol is added at the same place as when using the *Add New* command.

Editors > HMSC Editor

Adds a HMSC diagram symbol to the Organizer view, and starts the HMSC Editor. The symbol is added at the same place as when using the *Add New* command.

Editors > MSC Editor

Adds an MSC diagram symbol to the Organizer view, and starts the MSC Editor. The symbol is added at the same place as when using the *Add New* command.

Editors > OM Editor

Adds an Object Model diagram symbol to the Organizer view, and starts the OM Editor. The symbol is added at the same place as when using the *Add New* command.

Editors > SDL Editor

Adds an SDL System diagram symbol to the Organizer view, and starts the SDL Editor. The symbol is added at the same place as when using the *Add New* command.

Editors > State Chart Editor

Adds a State Chart diagram symbol to the Organizer view, and starts the State Chart Editor. The symbol is added at the same place as when using the *Add New* command.

Editors > Text Editor

Adds a Plain Text document symbol to the Organizer view, and starts the Text Editor. The symbol is added at the same place as when using the *Add New* command.

Editors > TTCN Browser

Adds a TTCN module symbol to the Organizer view, and starts the TTCN Browser. The symbol is added at the same place as when using the *Add New* command.

SDL > Type Viewer

This menu choice starts the Type Viewer. It is dimmed if there is no SDL diagram in the Organizer, or if the preference SDT*StartInformationServer is set to false. Only one instance of the Type Viewer exists. If the Type Viewer has already been started, its window is raised.

If a Referenced Diagram Type icon, an Instance Diagram icon or a Dashed diagram icon is selected in the Organizer, the Type Viewer selects the corresponding symbol when this menu choice is used.

The Type Viewer is described in chapter 46, *The SDL Type Viewer*.

SDL > Coverage Viewer

This menu choice starts a Coverage Viewer. A new instance of the Coverage Viewer is started each time this command is selected.

The Coverage Viewer is described in chapter 48, *The SDL Coverage Viewer*.

SDL > Index Viewer

This menu choice starts an Index Viewer. A new instance of the Index Viewer is started each time this command is selected.

The Index Viewer is described in chapter 47, *The SDL Index Viewer*.

SDL > Simulator UI

This menu choice starts a new, empty Simulator UI. Several Simulator UI's may exist at the same time.

The Simulator UI is described in "Graphical User Interface" on page 2130 in chapter 50, *The SDL Simulator*.

SDL > Validator UI

This menu choice starts a new, empty Validator UI. Several Validator UI's may exist at the same time.

The Validator UI is described in "Graphical User Interface" on page 2281 in chapter 53, *The SDL Validator*.

SDL > Target Tester UI

This menu choice starts a new, empty SDL Target Tester UI. The SDL Target Tester UI is described in "Graphical User Interface" on page 3562 in chapter 68, *The SDL Target Tester*.

TTCN > Find Table

This menu choice invokes the find table operation on a selected TTCN system. **In Windows**, this functionality is not available and the menu choice is dimmed.

For more information about the find table operation **on UNIX**, see "Finding Tables" on page 1196 in chapter 27, *Analyzing TTCN Documents (on UNIX)*. For more information about finding tables **in Windows**, see "Finding and Sorting Tables" on page 1260 in chapter 31, *Editing TTCN Documents (in Windows)*.

TTCN > Access

This menu choice starts a TTCN Access application for a selected TTCN document/system.

> **Note:**
> This operation is only available **on UNIX**.

For more information, see chapter 22, *TTCN Access*.

TTCN > Simulator UI

This menu choice starts a TTCN simulator UI.

For more information about this, see chapter 29, *The TTCN-SDL Co-Simulator (on UNIX)* or chapter 34, *The TTCN-SDL Co-Simulator (in Windows)*.

UML Suite

This menu choice starts the Telelogic Tau UML Suite. This menu choice is dimmed if the UML Suite is not installed on your system.

Preference Manager

This menu choice starts the Preference Manager. Only one instance of the Preference Manager exists. If the Preference Manager has already been started, its window is raised.

The Preference Manager is described in chapter 3, *The Preference Manager*.

Bookmarks Menu

The *Bookmarks* menu contains the following menu choices:

- *Add Bookmark*
- *Edit Bookmarks*
- *More Bookmarks*

Add Bookmark

This menu choice opens a dialog where a new bookmark can be created. The information needed is the *location* (a URL or a SDTREF), and a *name* for the bookmark. If desired, a *systemfile* can be specified that will be loaded together with the SDTREF.

Edit Bookmarks

Edit Bookmarks will open a dialog with a list with all bookmarks. After selecting a bookmark the buttons will have the following effect:

- *OK* - all changes made willbe saved and the dialog will close.
- *Open* - the selected bookmark will be opened
- *Edit* - provides a dialog with the information form the selected bookmark, where the information can be edited.
- *Remove* - removes the selected bookmark from the list.
- *Cancel* - all alteration made will be disregarded and the dialog will be closed.

More Bookmarks

This choice will appear if more than 25 bookmarks are present. The dialog appearing is the same as in the *Edit Bookmark* menu choice. Selecting the desired bookmark and clicking on the *Open* button will open the bookmark.

Help Menu

For more information about *Help* menus in Telelogic Tau, see "Help Menu" on page 15 in chapter 1, *User Interface and Basic Operations*. Two of the Organizer *Help* menu choices are described in more detail below.

About All

This menu choice starts an operation that presents version information about the individual tools in your SDL Suite configuration. The information is presented in the Organizer Log window. The produced information might look like this:

```
About All. Version information:
Help Tool              Version 4.5.0
Link Manager           Version 4.5.0
MSC Editor             Version 4.5.0
OM InfoServer          Version 4.5.0
OM/SC/HMSC/DP Editor   Version 4.5.0
Organizer              Version 4.5.0
Preference Manager     Version 4.5.0
SDL Coverage Viewer    Version 4.5.0
SDL Editor             Version 4.5.0
SDL Index Viewer       Version 4.5.0
SDL Type Viewer        Version 4.5.0
SDT Welcome Window     Version 4.5.0
Text Editor            Version 4.5.0

About All. Additional version and kernel
information:
Information Server version 4.5.0
SDT Analyzer
SDT Analyzer 4.5.0
SDT CPP2SDL 4.5.0
ASN.1 Analyzer 4.5.0
SDL Targeting Expert: Version 4.5.0

Simulation kernel:
   2 lrwxrwxrwx   1 lat      develop       66 May  5
16:05 sctworld.o
RealTimeSimulation kernel:
   2 lrwxrwxrwx   1 lat      develop       68 May  5
16:05 sctworld.o
...
```

License Information

Opens a dialog with license information for all tools in Telelogic Tau.

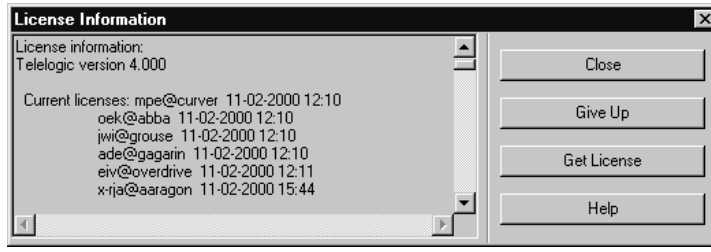

Figure 52: The License Information dialog

The dialog contains the following:

- The names of the tools that are included in Telelogic Tau and that are separately licensed.

  Each name consists of the prefix SDT or ITEX, followed by a hyphen and an abbreviated name for the tool. See the table below for a reference to these abbreviations.

Abbreviation	**Telelogic Tau Tool(s)**
Telelogic	Organizer
SDT-Base	SDL Editor, SDL Analyzer and SDL Viewers.
SDT-OME	Object Model Editor and State Chart Editor
SDT-MSCE	Message Sequence Chart Editor and High-Level Message Sequence Chart Editor
SDT-Cbasic	Cbasic SDL to C Compiler
SDT-Cadvanced	Cadvanced SDL to C Compiler
SDT-Cmicro	Cmicro SDL to C Compiler
SDT-Cmicro-Bodybuilder	Cmicro BodyBuilder
SDT-Cmicro-Tester	SDL Target Tester UI
SDT-X[a] **(UNIX only)**	<Configuration dependent>

Abbreviation	Telelogic Tau Tool(s)
SDT-Validator	Validator Library
SDT-Simulator	Simulator Library
SDT-Application	Application Library
SDT-Performance	Performance Simulation Library
SDT-TTCN-Link	Interactive link to ITEX
ITEX-Base	TTCN Browser, TTCN Table Editor and TTCN Analyzer
ITEX-Access	ITEX Access API Library
ITEX-Simulator	ITEX Simulator
ITEX-C-Code-Generator	ITEX C Code Generator

a. The SDT-X is a generic name that allows to introduce new code generators that are under development. In “normal” installations, it has no meaning.

- **On UNIX**, beneath each tool, the following information is displayed:
  - Current licenses, along with the name of the host computer, the user identity and the time the license was checked out
  - Remaining available licenses
  - Total available licenses
  - If a license has expired, the information would be replaced by:
    ```
    SDT-<ToolName>
      Expired: <Date>
    ```
- A *Give Up License* button which allows you to release the interactive licenses in your current SDL suite session.

  This will disable licensed tools without closing them, making the licenses available to other users. You can reclaim these licenses at a later time, if there are corresponding licenses available, by using the *Get License* button, allowing you to continue working in the licensed applications.

- A *Get License* button which allows you to reclaim licenses released by either the use of the *Give Up License* button or the automatic release of licenses.

  Licenses will be automatically released if the idle time limit, set in the preference LicenseTimeout, is exceeded. For more information, see "LicenseTimeout" on page 238 in chapter 3, *The Preference Manager*.

  The reclamation of licenses is limited in the sense that it will only attempt to reclaim lost licenses once, implying that you will have to save your work in and restart the tools for which licenses could not be obtained.

Shortcuts

There are several ways to invoke an operation in the Organizer. An operation is usually available in the menu bar and in at least one pop-up menu. Menus can be operated via the keyboard (keyboard navigation). Several operations are also available in the tool bar or via keyboard accelerators. For instance, the Print operation can be invoked by:

- Choosing the menu choice in the *File* menu in the menu bar with the mouse.
- Choosing the menu choice in the *File* menu in the menu bar with keyboard navigation. (`<Alt+F> P`)
- Using the keyboard accelerator `<Ctrl+P>`.
- Pressing the *Print* quick button.
- Choosing the menu choice in the *File* sub menu of the background pop-up menu. (The *Print* menu choice is also available in several other pop-up menus.)

The sections below are "Pop-Up Menus" on page 166, "Keyboard Accelerators" on page 174 and "Quick Buttons" on page 176.

Pop-Up Menus

This section describes pop-up menus for different types of icons. The operations available depend on the type of file that icon represents.

The following tables lists the menu choices in the pop-up menus and a reference to the corresponding menu choice in the menu bar.

On SDL Diagrams (but not SDL Overview Diagrams)

Print Selected	"Print > Selected" on page 69
Set Directories	"Set Directories" on page 70
Edit > Edit	"Edit" on page 85
Edit > Remove	"Remove" on page 90
Edit > Connect	"Connect" on page 91
Edit > Disconnect	"Disconnect" on page 95

Edit > CM Group	“Configuration > Group File” on page 96
Edit > CM Update	“Configuration > Update” on page 97
Edit > Associate	“Associate” on page 97
Edit > Update Headings	“Update Headings” on page 99
View > Expand	“Expand” on page 102
View > Expand Substructure	“Expand Substructure” on page 103
View > Collapse	“Collapse” on page 103
View > Show Sub Symbols	“Show Sub Symbols” on page 104
View > Hide	“Hide” on page 105
Generate > Analyze	“Analyze” on page 111
Generate > Make	“Make” on page 119
Generate > Stop Analyze/Make	“Stop Analyze/Make (UNIX only)” on page 126
Generate > Generate SDL Overview	“SDL Overview” on page 127
Generate > Convert to PR	“Convert to PR/MP” on page 131
Generate > Convert to GR	“Convert to GR” on page 133
Generate > Edit Separation	“Edit Separation” on page 136
Generate > Dependencies	“Dependencies” on page 137
Search	“Search” on page 140
Change Bars	“Change Bars” on page 145
Compare SDL Diagrams	“Compare SDL Diagrams” on page 146
Merge SDL Diagrams	“Merge SDL Diagrams” on page 149
Create Endpoint	“Link > Create Endpoint” on page 140
Traverse Link	“Link > Traverse” on page 140
Clear Endpoint	“Link > Clear Endpoint” on page 140

Type Viewer	**(On SDL type diagrams only)** "SDL > Type Viewer" on page 158. Opens the Type Viewer with the type diagram selected.

On MSC and SDL Overview Diagrams

Print Selected	"Print > Selected" on page 69
Set Directories	"Set Directories" on page 70
Edit > Edit	"Edit" on page 85
Edit > Remove	"Remove" on page 90
Edit > Connect	"Connect" on page 91
Edit > Disconnect	"Disconnect" on page 95
Edit > CM Group	"Configuration > Group File" on page 96
Edit > CM Update	"Configuration > Update" on page 97
Edit > Associate	"Associate" on page 97
Hide	"Hide" on page 105
Dependencies	"Dependencies" on page 137
Search	"Search" on page 140
Change Bars	"Change Bars" on page 145
Create Endpoint	"Link > Create Endpoint" on page 140
Traverse Link	"Link > Traverse" on page 140
Clear Endpoint	"Link > Clear Endpoint" on page 140

On Instance and Dashed Diagrams

Edit	"Edit" on page 85.
Type Viewer	"SDL > Type Viewer" on page 158. Opens the Type Viewer with the diagram selected.
Hide	"Hide" on page 105

On Pages

Print Selected	"Print > Selected" on page 69
Edit	"Edit" on page 85.
Hide	"Hide" on page 105

On Associations

Associations have the same pop-up menu as the associated document. All menu choices operate on the associated document, except *Hide* and *Remove* that operate on the association symbol itself.

On HMSC, Object Model and State Chart Diagrams

Print Selected	"Print > Selected" on page 69
Set Directories	"Set Directories" on page 70
Edit > Edit	"Edit" on page 85
Edit > Remove	"Remove" on page 90
Edit > Connect	"Connect" on page 91
Edit > Disconnect	"Disconnect" on page 95
Edit > CM Group	"Configuration > Group File" on page 96
Edit > CM Update	"Configuration > Update" on page 97
Edit > Associate	"Associate" on page 97
Edit > Update Headings	"Update Headings" on page 99
View > Expand	"Expand" on page 102
View > Expand Substructure	"Expand Substructure" on page 103
View > Collapse	"Collapse" on page 103
View > Show Sub Symbols	"Show Sub Symbols" on page 104
View > Hide	"Hide" on page 105
Dependencies	"Dependencies" on page 137
Search	"Search" on page 140

Change Bars	“Change Bars” on page 145
Create Endpoint	“Link > Create Endpoint” on page 140
Traverse Link	“Link > Traverse” on page 140
Clear Endpoint	“Link > Clear Endpoint” on page 140

On Modules

Print Selected	“Print > Selected” on page 69
Edit	“Edit” on page 85
Remove	“Remove” on page 90
CM Group	“Configuration > Group File” on page 96
CM Update	“Configuration > Update” on page 97
View > Expand	“Expand” on page 102
View > Expand Substructure	“Expand Substructure” on page 103
View > Collapse	“Collapse” on page 103
View > Show Sub Symbols	“Show Sub Symbols” on page 104
Merge ASN.1	“Merge ASN.1” on page 138
Search	“Search” on page 140
Change Bars	“Change Bars” on page 145
Compare SDL Diagrams	“Compare SDL Diagrams” on page 146
Merge SDL Diagrams	“Merge SDL Diagrams” on page 149

On Chapters

Print Selected	“Print > Selected” on page 69
Edit	“Edit” on page 85
Remove	“Remove” on page 90

CM Group	"Configuration > Group File" on page 96
Search	"Search" on page 140
Compare SDL Diagrams	"Compare SDL Diagrams" on page 146

On More Symbols

Show Sub Symbols	"Show Sub Symbols" on page 104

On System File

CM Group	"Configuration > Group File" on page 96
CM Update	"Configuration > Update" on page 97

On Link File

Link Manager	"Link > Link Manager" on page 140

On Source and Target Directories

Set Directories	"Set Directories" on page 70

On the Background

Expand Substructure	"Expand Substructure" on page 103
Collapse	"Collapse" on page 103
View Options	"View Options" on page 105
Set Scale	"Set Scale" on page 110
File > New	"New" on page 58
File > Open	"Open" on page 59
File > Save	"Save" on page 61
File > Save As	"Save As" on page 64

File > Print All	"Print > All" on page 69
File > Set Directories	"Set Directories" on page 70
File > PC Drives	"PC Drives" on page 72
File > Compare System	"Compare System" on page 73
File > Merge System	"Merge System" on page 73
File > Import SDL	"Import SDL" on page 79
File > Exit	"Exit" on page 83
Edit > Add New	"Add New" on page 88
Edit > Add Existing	"Add Existing" on page 89
Edit > Paste As	"Paste As" on page 98
Edit > Go to Source	"Go To Source" on page 99
Convert to GR	"Convert to GR" on page 133
Tools > Organizer Log	"Organizer Log" on page 140
Tools > Link Manager	"Link > Link Manager" on page 140
Tools > Search	"Search" on page 140
Tools > Change Bars	"Change Bars" on page 145
Tools > Compare SDL Diagrams	"Compare SDL Diagrams" on page 146
Tools > Merge SDL Diagrams	"Merge SDL Diagrams" on page 149
Tools > Type Viewer	"SDL > Type Viewer" on page 158
Tools > Coverage Viewer	"SDL > Coverage Viewer" on page 159
Tools > Index Viewer	"SDL > Index Viewer" on page 159
Tools > Simulator UI	"SDL > Simulator UI" on page 159
Tools > Validator UI	"SDL > Validator UI" on page 159
Tools > Target Tester UI	"SDL > Target Tester UI" on page 159
Tools > UML Suite	"UML Suite" on page 160
Tools > Preference Manager	"Preference Manager" on page 160

Editors > Deployment Editor	"Editors > Deployment Editor" on page 157
Editors > HMSC Editor	"Editors > HMSC Editor" on page 157
Editors > MSC Editor	"Editors > MSC Editor" on page 158
Editors > OM Editor	"Editors > OM Editor" on page 158
Editors > SDL Editor	"Editors > SDL Editor" on page 158
Editors > State Chart Editor	"Editors > State Chart Editor" on page 158
Editors > Text Editor	"Editors > Text Editor" on page 158
Editors > TTCN Browser	"Editors > TTCN Browser" on page 158
Help > Contents *Help > On Organizer* *Help > On Shortcuts* *Help > New Features* *Help > Latest News* *Help > Index* *Help > Search* *Help > Telelogic Home Page* *Help > Help Desk* *Help > About Organizer*	For more information about *Help* menu choices, see "Help Menu" on page 15 in chapter 1, *User Interface and Basic Operations*.
Help > License Information	"License Information" on page 162

Keyboard Accelerators

Apart from the general keyboard accelerators, as described in "Keyboard Accelerators" on page 35, the following accelerators can be used in the Organizer:

Accelerator	Reference to corresponding command or quick button
Ctrl+A	"Add New" on page 88
Ctrl+B	"Connect" on page 91
Ctrl+E	"Edit" on page 85
Ctrl+I	"Import SDL" on page 79
Ctrl+K	"Stop Analyze/Make (UNIX only)" on page 126
Ctrl+M	"Make" on page 119
Ctrl+R	"Disconnect" on page 95
Ctrl+T	Toggles between *Analyze*/*Make* (default) and *Full Analyze*/*Full Make*. A text is presented in the message area to confirm that <Ctrl+T> has been pressed: Either "Organizer will now use full analyze/make" or "Organizer will now use normal analyze/make". For more information, see "Normal versus full analyze (make)" on page 113.
Ctrl+0 (zero)	"Set Directories" on page 70
Ctrl+1	"Organizer Log" on page 140
Ctrl+2	"Add Existing" on page 89
Ctrl+3	"Associate" on page 97
Ctrl+4	"Analyze" on page 111
Ctrl+5	"SDL Overview" on page 127
Ctrl+6	"SDL > Type Viewer" on page 158
Ctrl+7	"SDL > Coverage Viewer" on page 159
Ctrl+8	"SDL > Index Viewer" on page 159

Accelerator	Reference to corresponding command or quick button
`Ctrl+9`	“Preference Manager” on page 160
`Del`	“Remove” on page 90. For documents that can not be removed (non-root SDL diagrams), see “Disconnect” on page 95.
`arrow up`	Select the symbol one step up (move the selection)
`Shift+arrow down`	“Move Up” on page 178.
`Ctrl+Shift+ arrow up`	Place the selected symbol first in the Organizer structure.
`arrow down`	Select the symbol one step down (move the selection)
`Shift+arrow down`	“Move Down” on page 178
`Ctrl+Shift+ arrow down`	Place the selected symbol last in the Organizer structure.

Quick Buttons

The following quick buttons are special to the Organizer's Main window. The general quick buttons are described in "General Quick-Buttons" on page 24. Each quick button in the Organizer has a preference parameter that specifies if the button is shown or hidden; see "Organizer Preferences" on page 240. As default, all quick buttons except *New* and *Add Existing* are shown.

New
Creates a new system; see "New" on page 58.

Save
Performs a silent *Save All* of all diagrams including the system file. Does not bring up the *Save* dialog, except if diagrams are modified and unconnected, or no system file exists. Corresponds to the *Save All* button in the Save dialog; see "Save" on page 61.

Print
Brings up the *Print* dialog, see chapter 5, *Printing Documents and Diagrams*.

Analyze
Analyzes the part of the system defined by the selected diagram or the whole system. Brings up a *Save Before* dialog if any diagram is modified and not saved. Does not bring up the *Analyze* dialog. Corresponds to the *Analyze* button in the Analyze dialog. The current Analyze options are used. The same restrictions and checks apply as described in "Analyze" on page 111.

Make
Makes the part of the system defined by the selected diagram or the whole system, i.e. generates code, compiles and links it. Does not bring up the *Make* dialog. Corresponds to the *Make* button in the Make dialog. The current Analyze and Make options are used. The same restrictions and checks apply as described in "Make" on page 119.

Generate SDL Overview
Generates an SDL overview diagram for the selected diagram or the whole system. Does not bring up the *SDL Overview* dialog. Corresponds to the *Generate* button in the *Generate SDL Overview* dialog. The current *Generate SDL Overview* options are used. The same restrictions and checks apply as described in "SDL Overview" on page 127.

Simulate
Generates a simulator for the part of the system defined by the selected diagram or the whole system, and starts a Simulator UI (see "Simulator/Validator UI Started from Quick Buttons" on page 178). Does not bring up the *Make* dialog if the simulator needs to be regenerated. If a simulation kernel has not been selected in this dialog, the default simulation kernel is used. The default simulation kernel is determined by the preference Organizer*DefaultSimulator. The default value is "SCTADEBCOM" which is the same as using the standard kernel *Simulation* in the Make dialog. The same restrictions and checks apply as described in "Make" on page 119.

Validate
Generates a validator for the part of the system defined by the selected diagram or the whole system, and starts a Validator UI (see "Simulator/Validator UI Started from Quick Buttons" on page 178). Does not bring up the *Make* dialog if the validator needs to be regenerated. If a validation kernel has not been selected in this dialog, the default validation kernel is used. The default validation kernel is determined by the preference Organizer*DefaultValidator. The default value is "SCTAVALIDATOR" which is the same as using the standard kernel *Validation* in the Make dialog. The same restrictions and checks apply as described in "Make" on page 119.

Generate Cross References
Generates a cross reference file for the current system and presents the information in the Index Viewer. An index symbol is added in the Organizer view. Does not bring up the *Analyze* dialog. Corresponds to setting the *Generate a cross reference file* option in this dialog. The same restrictions and checks apply as described in "Analyze" on page 111 and "Semantic analysis" on page 114.

Add New
Adds a new document as root; see "Add New" on page 88.

Add Existing
Adds an existing document as root; see "Add Existing" on page 89.

Move Down
Moves the selected symbol one step down in the Organizer structure. For more information about moving symbols, see "Ordering" on page 54.
Moves down the selected page. For more information about moving pages, see "Move down" on page 1969 in chapter 44, *Using the SDL Editor* and "Move down" on page 1636 in chapter 40, *Using Diagram Editors*.

Move Up
Moves the selected symbol one step up in the Organizer structure. For more information about moving symbols, see "Ordering" on page 54.
Moves up the selected page. For more information about moving pages, see "Move up" on page 1968 in chapter 44, *Using the SDL Editor* and "Move up" on page 1636 in chapter 40, *Using Diagram Editors*.

Toggle Pages
Shows or hides page symbols below diagram symbols in the Organizer structure.

Organizer Log
Opens the Organizer Log window; see "Organizer Log" on page 140.

Simulator/Validator UI Started from Quick Buttons

When the Simulate quick button is used and if the generated simulator already was loaded in a Simulator UI, the same Simulator UI will be re-used for the newly generated simulator. If no Simulator UI is active, a new Simulator UI will be started. However, if there already are other started Simulator UI's, the user has the choice of selecting one of these to load the generated simulator in.

The following dialog is opened:

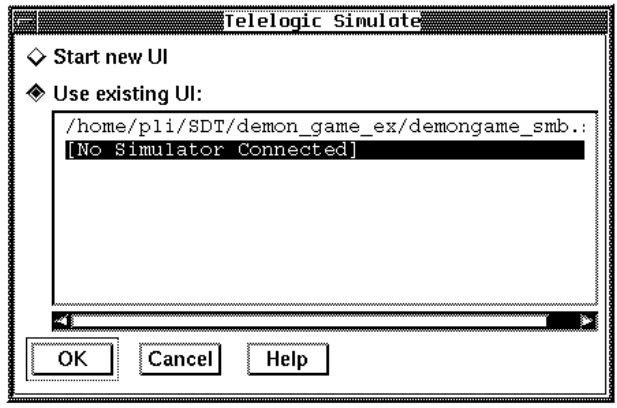

Figure 53: The Simulator/Validator dialog

- *Start new UI*

  Starts a new Simulator UI and loads the simulator.

- *Use existing UI*

  From the list box, one of the active Simulator UI's can be selected to load the simulator in. If a simulator already was running in the selected Simulator UI, that simulator will be terminated.

The same applies also for a validator and the Validator UI.

Organizer Log Window

The Organizer log window is a sub window to the main window. It works as a console for the Telelogic Tau tools. The window can be visible (raised or iconified) or not visible on the screen. All log information is output to this window independently of whether the window is visible or not.

The window is used in the following situations:

- To log information from the Analyze and Make process.
- To log the activities when an SDL system is imported.
- When files are checked during the *Open* command. Inconsistent files are reported.

- To show status information from the Generate tools.
- Other tools, such as ITEX and tools started with dynamic menus, may also use the window to output textual information.

The window is opened or raised when the menu choice *Organizer Log* is selected from the *Tools* menu, or when something is written to the log. The preference ShowLogLevel controls which output to the window should cause a raise of the window (see chapter 3, *The Preference Manager*).

There is only one Organizer Log window. The Organizer main window is not locked for user input when the Organizer Log window is visible.

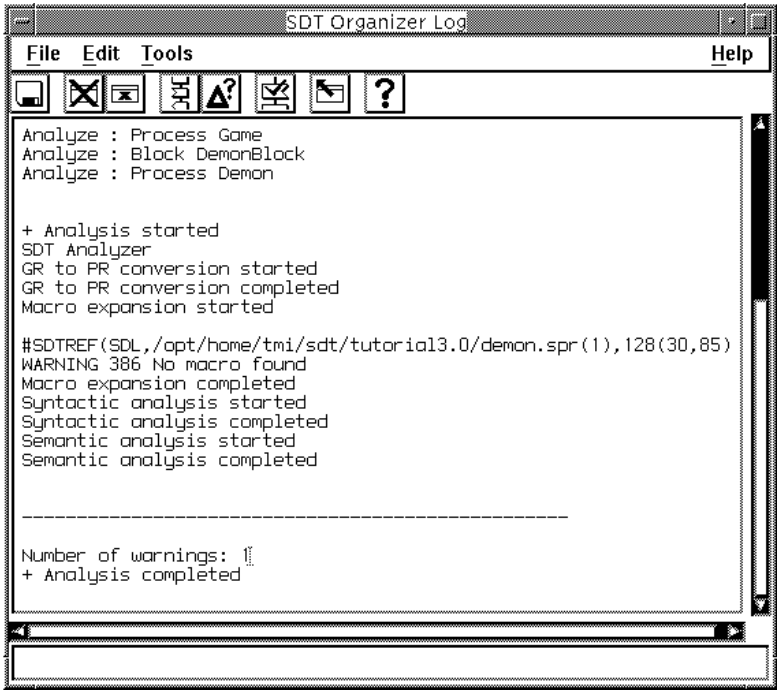

Figure 54: The Organizer Log window

Quick Buttons

The following quick buttons are special to the Organizer Log window. The general quick buttons are described in "General Quick-Buttons" on page 24. Each quick button in the Organizer Log window has a preference parameter that specifies if the button is shown or hidden; see "Organizer Preferences" on page 240 in chapter 3, *The Preference Manager*. All quick buttons in the Organizer Log are shown by default.

Close
Exactly the same as *Close* in the *File* menu.

Clear Log
Exactly the same as *Clear Log* in the *Edit* menu; see "Clear Log" on page 182.

Navigate Down
Move to the next error visible in the Organizer Log. See "Log Options" on page 182 for details.

Navigate Up
Move to the previous error visible in the Organizer Log. See "Log Options" on page 182 for details.

Show Error
Exactly the same as *Show Error* in the *View* menu; see "Show Error" on page 183.

Help on Error
Exactly the same as *Help on Error* in the *View* menu; see "Help on Error" on page 183.

Analyze
Exactly the same as the *Analyze* quick button in the Organizer Main window; see "Analyze" on page 176.

Show Organizer
Exactly the same as *Show Organizer* in the *Tools* menu; see "Show Organizer" on page 182.

File Menu

See "File Menu" on page 8 in chapter 1, *User Interface and Basic Operations*.

Edit Menu

The following menu choices are available in the Organizer log window *Edit* menu:

Copy

This menu choice copies the selected text to the clipboard buffer.

Select All

This menu choice selects all text in the Organizer Log window.

Clear Log

This menu choice clears all text in the Organizer Log window.

Log Options

This menu choice lets you decide how error, warning and information messages from the Analyzer should be handled in the Organizer Log:

- The Filter text field can be used to specify warning messages that should not be shown.
- The navigation level decides if the navigate up and down quick buttons should navigate through:
  - Error messages
  - Error and warning messages
  - All messages

The default value for the filter is taken from the preference Organizer*LogFilter. The default value for the navigation level is taken from the preference Organzier*LogNavigationLevel.

Tools Menu

The following menu choices are available in the *Tools* menu of the Organizer log window:

Show Organizer

This menu choice raises the Organizer's Main window.

Show Error

This menu choice opens an editor and shows an object/text that corresponds to a selected error/warning. If no text in the log is selected, the menu choice operates on the first error/warning in the log, which becomes selected. If this menu choice is invoked several times without changing the selection manually, the selection is updated automatically to point to the next error/warning before opening the editor and selecting the corresponding object/text.

Help on Error

This menu choice activates the Help tool on the selected error/warning or the next found error/warning. The selection is not updated automatically when selecting this command repeated times (as opposed to *Show Error*, above).

System File

The Organizer maintains a system file that contains a description of all files that are included in the system, together with all settings the user has made to this particular system. The default file name extension for a system file is `.sdt`.

In addition, a system window state file contains information about window positions and sizes, and it uses the extension `.sdt.state`.

Contents of the System File

The contents of the system file summarizes the information that the Organizer manages:

- Source Directory and Target Directory.
- Link file reference (used by the Link Manager) and information about endpoints in the Organizer view.
- All document names are listed, both with logical names and file names. The fact that document names are present means that the Organizer will have a robust recovery when files are manipulated outside the Organizer.
- Associations and dependencies are stored as indexes to other documents for a document.
- The names of all chapters and modules are listed.
- The names of all Control Unit files.

The system file may contain a mapping of directory paths on Windows and UNIX to allow the system to be accessed on both platforms. See "Windows and UNIX File Compatibility" on page 209 for more information.

A number of options are also stored in the system file, but only if they differ from the values in the preference file or if the preference Organizer*AllPreferences is on. The options are:

- Window options as set in the *View Options* and *Set Scale* dialogs
- Analyze options as set in the *Analyze* dialog
- Make options as set in the *Make* dialog
- Print options as set in the *Print > Selected* dialog

In addition, information about collapsed, hidden and selected documents are saved to enable the Organizer to open with the same appearance as when the system file was saved. However, simply changing which documents are collapsed, hidden and selected will not mark the system file as "dirty" in the Organizer.

Format of the System File

The system file is a line-oriented, human-readable text file. The system file is divided into sections. Each section contains lines formatted in the same way. Any lines before the first section are ignored, but saved for the next system file save. This means that it is possible to have user defined comments in the beginning of the file.

> **Caution!**
>
> The sections that describe the diagram and document structure should not be modified by the user unless absolutely necessary. Modifying these sections in a way that errors or inconsistencies are introduced may corrupt the system file.

The file has the following format:

```
<file> ::= <comments> SDT-SYSTEM-4.5(<drives> | $)
           (<dir> | $) (<links> | $)
           (<diagram> | $) (<view> | $) (<analys> | $)
           (<cpp2sdl> | $) (<make> | $) (print | $)

<comments> ::= <comment>*
<comment> ::= Any line of ASCII characters not
beginning with "SDT-SYSTEM-".

<drives> ::= [DRIVES] <drive>*
<drive> ::= <Windows dir> <UNIX dir>

<dir> ::= [SOURCE-TARGET-DIRECTORY] <option>*
<links> ::= [LINKS] <option>*

<diagram> ::= [DIAGRAMS] <dia>*
<dia> ::= <indent> <type> <name>
          (<path> | <fileName> | %unconnected%)
          <viewState> <timeStamp> <separateName>
          <associations> <dependencies>

<view> ::= [VIEWOPTIONS] <option>*
<analys> ::= [ANALYSEROPTIONS] <option>*
<cpp2sdl> ::= [CPP2SDLOPTIONS] <option>*
<make> ::= [MAKEOPTIONS] <option>*
```

<print> ::= [PRINTOPTIONS] <option>*

<Windows dir> ::= <string>
An Windows path. This is either a drive letter and a colon (e.g. C:) or a full path (e.g. C:\SDL). UNC paths can be used (e.g. \\MYHOST\SDL). If you include a trailing backslash you must also include a trailing slash in the corresponding <UNIX dir>.

<UNIX dir> ::= <string>
A UNIX directory path corresponding to the <Windows dir>. If you include a trailing slash you must also include a trailing backslash in the corresponding <Windows dir>.

<indent> ::= <int>
The indentation level. 0 is root, -1 is used for chapters.

<type> ::= <int>
Type of document, module or chapter. If the diagram has some kind of virtuality, the value is factored with a number with the base 100. The value corresponds to the kind of virtuality. These numbers correspond to an enumerated value found in the file sdt.h. If the diagram is an instance, a value of 1000 will be added to the type.

<name> ::= <string>
Logical name of diagram or document.

<path> ::= <string>
File path including a directory.

<fileName> ::= <string>
File name.

<viewState> ::= <integer>
Viewing state, consisting of four weighted booleans for separation, expanded, shown and selected states:
Separation + 2*NotExpanded + 4*NotShown + 8*Selected.

<timeStamp> ::= <integer>
Last time the file was modified, and that the Organizer was aware of. When the Organizer reads the system file, only files with modification dates later than the time stamp are checked for correctness.

<separateName> ::= <string>
Name of separation. Only applicable for units that are separately analyzed.

<associations> ::= <string>
A string of space separated values referencing associated documents, e.g. “1 4 5”. The values index documents in the DIAGRAMS section.

<dependencies> ::= <string>

A string of space separated values referencing documents that this document is depending on, e.g. “1 4 5”. The values index documents in the DIAGRAMS section.

```
<option> ::= (<option-name> = <option-value>)
<option-name> ::= <string>
```
Any option found in the named dialogs.

```
<option-value> ::= <string>
```
Any valid value bound to an option.

If a section is missing entirely from the system file, a warning will be logged, except for the `cpp2sdl` section which is optional and only useful in batch. If no recognizable sections could be found, the file is not a valid system file.

When loading system files, warnings will be registered in the Organizer Log window if any non-recognized option is encountered, but not if an option never was initialized.

The Drives Section

The purpose of the `DRIVES` section in the system file is to achieve file compatibility between UNIX and Windows systems. It specifies the *drive table*; a mapping between Windows and UNIX directory paths. See “Windows and UNIX File Compatibility” on page 209 for more information.

The path format of the current platform is used for directory paths stored in the system file: UNIX path names on UNIX systems (i.e. starting with a slash ‘/’ and directories separated by slashes), and Windows path names on Windows systems (i.e. starting with a drive letter “x:” and directories separated by backslashes ‘\’, or using the UNC format “\\<host>\file”).

Diagrams with file specifications in an incorrect format are marked as Invalid in the Organizer.

When reading the system file on Windows systems, file specifications in UNIX format are converted to Windows format using the mapping in the `DRIVES` section, if possible, including converting slashes to backslashes. When the system file is saved, the file specifications are saved in Windows format, i.e. they are not converted back to UNIX format.

On UNIX systems, file specifications in Windows format are converted to UNIX format using the mapping in the `DRIVES` section, if possible,

including converting backslashes to slashes. When the system file is saved, the file specifications are saved in UNIX format, i.e. they are not converted back to Windows format.

Example 3 ----------

The `DRIVES` section of the system file looks like this:

```
[DRIVES]
C:\TEMP /tmp
\\MYHOST\STORAGE /home/user
```

The file specification `/tmp/a.ssy` in the system file is converted to `C:\TEMP\a.ssy`.

The file specification `/home/user/mydir/a.ssy` is converted to `\\MYHOST\STORAGE\mydir\a.ssy`.

The file specification `/usr/local/dir/a.ssy` is converted to `\usr\local\dir\a.ssy`. Since no drive or host name could be matched, the file will not be found and the diagram will be marked as Invalid in the Organizer.

Options in the System File

A number of sections contain options for the Organizer, representing values that can be set in the Organizer dialogs. If an option in the file is not recognized by the Organizer, it will be ignored, and if an option is not included in the file, a default preference value will be used. A few of the options have no corresponding preference parameters.

The following tables list the Organizer options that are saved in the system file. The user should not normally need to know or change these options in the system file. However, when running the SDL suite in batch mode **on UNIX**, it may be useful to change some options by editing a system file to be submitted as input to `sdtbatch`. See "Batch Facilities" on page 203 for more information.

The options must appear in the correct section of the system file, but the ordering of options within a section is not important. The options are stored in the system file according to the format:

```
NameOfOption=Value
```

The option names and option values are case insensitive.

Possible and allowed values are not specified; the user should run the Preference Manager in order to obtain a reference to the permitted values.

SOURCE-TARGET-DIRECTORY Section

Option	Default	Corresponding Preference
AbsolutePath	false	*Organizer*AbsolutePath*
SourceDirectory	""	*Organizer*SourceDirectory*
TargetDirectory	""	*Organizer*TargetDirectory*

VIEWOPTIONS Section

Option	Default	Corresponding Preference
Scale	100	*SDT*Scale*
ShowDashed	true	*Organizer*ShowDashed*
ShowDependencies	true	*Organizer*ShowDependencies*
ShowDirectories	true	*Organizer*ShowDirectories*
ShowFileNames	true	*Organizer*ShowFileName*
ShowGroups	true	*Organizer*ShowGroups*
ShowInstances	true	*Organizer*ShowInstances*
ShowLinkFile	false	*Organizer*ShowLinkFile*
ShowLinks	true	*Organizer*ShowLinks*
ShowLongMenus	true	*Organizer*ShowLongMenus*
ShowPages	false	*Organizer*ShowPages*
ShowPermissions	true	*Organizer*ShowPermissions*
ShowSeparators	true	*Organizer*ShowSeparators*
ShowStatusBar	true	*Organizer*Statusbar*
ShowSystemFile	true	*Organizer*ShowSystemFile*

Option	Default	Corresponding Preference
ShowToolBar	true	*Organizer*Toolbar*
ShowTypeName	false	*Organizer*ShowTypeName*
ShowVirtuality	true	*Organizer*ShowVirtuality*
TreeRepresentation	List	*Organizer*TreeRepresentation*

ANALYSEROPTIONS Section

Option	Default	Corresponding Preference
AllowImplicitTypeConv	false	*Organizer*AllowImplicitTypeConv*
ASN1KeywordFile	false	Not available
ASN1KeywordFileName	""	*Organizer*ASN1KeywordFileName*
ASN1Parameter	false	Not available
CaseSensitiveSDL	false	*SDT*CaseSensitive*
CoderBufferInSDL		*Organizer*CoderBufferInSDL*
EchoAnalyzerCommands	false	*Organizer*EchoAnalyzerCommands*
ErrorLimit	30	*Organizer*ErrorLimit*
ExpandPR	false	*Organizer*ExpandPR*
ExpressionLimit	0	*Organizer*ExpressionLimit*
Filter	false	*Organizer*FilterCommand*
FilterCommand	""	*Organizer*FilterCommand*
IgnoreHidden	true	Not available
IncludeOptionalFields	false	*Organizer*IncludeOptionalFields*
InstanceFile	false	Not available
MacroExpansion	false	*Organizer*MacroExpansion*
MissingAnswerValues-Control	true	*Organizer*MissingAnswerValues-Control*
MissingElseControl	true	*Organizer*MissingElseControl*

Option	Default	Corresponding Preference
OptionalParamControl	true	*Organizer*OptionalParamControl*
OutputControl	true	*Organizer*OutputControl*
ReferenceControl	true	*Organizer*ReferenceControl*
SemanticControl	true	*Organizer*SemanticControl*
SyntaxControl	true	*Organizer*SyntaxControl*
TerminateAnalyzer	false	*Organizer*TerminateAnalyzer*
TrailingParamControl	true	*Organizer*TrailingParamControl*
UpperCase	false	*Organizer*UpperCase*
UsageControl	true	*Organizer*UsageControl*
XRef	true	*Organizer*XRef*

CPP2SDLOPTIONS Section, optional section which can be used in batch mode

Option	Default	Corresponding Preference
InputLanguageC	false	Not available
InputLanguageCpp	true	Not available
InputLanguageBorland	false	Not available
InputLanguageMicrosoft	false	Not available
InputLanguageGNU	false	Not available
RTTI	false	Not available
RecognizeSDLsorts	false	Not available
ObjectSlicing	false	Not available
Preprocessor	""	Not available
UsePreprocessorOptions	false	Not available
PreprocessorOptions	""	Not available
PtrPrefix	ptr_	Not available

Option	Default	Corresponding Preference
KeywordPrefix	keyword_	Not available
ArrPrefix	arr_	Not available
IncompletePrefix	incomplete_	Not available
TplPrefix	tpl_	Not available
UscoreSuffix	uscore	Not available
NoAbsolutePath	false	Not available
GenerateCPPTypes	false	Not available
OptimizeClassPointers	false	Not available

MAKEOPTIONS Section

Option	Default	Corresponding Preference
Capitalization	AsDefined	*Organizer*Capitalisation*
CCompilerDriver	false	Not available
CompileAndLink	true	*Organizer*CompileAndLink*
FileNamePrefix	""	*Organizer*FileNamePrefix*
GenerateASN1Coder	false	*Organizer*GenerateASN1Coder*
GenerateCode	true	*Organizer*GenerateCode*
GenerateEnvFunctions	false	*Organizer*GenerateEnvFunctions*
GenerateEnvHeader	false	*Organizer*GenerateEnvHeader*
GenerateSDLCoder	false	*Organizer*GenerateSDLCoder*
GenerateSignalNumbers	false	*Organizer*GenerateSignalNumbers*
Kernel	"SCTADEBCOM"	*Organizer*Kernel*
MakefileMode	Generate	*Organizer*MakefileMode*
PrefixType	Full	*Organizer*PrefixType*
Separation	No	*Organizer*Separation*
StandardKernel	true	*Organizer*StandardKernel*

Option	Default	Corresponding Preference
TargetLanguage	Cbasic	*Organizer*TargetLanguage*
UserKernel	""	*Organizer*UserKernel*
UserMakefile	""	*Organizer*UserMakefile*
UserTemplate	""	*Organizer*UserTemplate*
XCodeGenerator	X	*Organizer*XCodeGenerator*

PRINTOPTIONS Section

Option	Default	Corresponding Preference
DestinationFormat	PSFile	*Print*DestinationFormat*
FirstPageNo	1	Not available
FrameMakerCommand	"imaker"	*Print*FrameMakerCommand*
MarginLeft	100	*Print*MarginLeft*
MarginLower	250	*Print*MarginLower*
MarginRight	100	*Print*MarginRight*
MarginUpper	420	*Print*MarginUpper*
PageMarkers	false	*Print*PageMarkers*
PaperFormat	A4	*Print*PaperFormat*
PrinterCommand	"lpr -h -r"	*Print*PrinterCommand*
PrinterFile	""	Not available
PrintFrom		Not available
PrintTo		Not available
PrintToFile	true	Not available

System Window State File

When the system file is saved, a second file is saved as well; the system window state file. This file contains information about window positions and sizes. If an editor window is not open when the system file is saved, the last known position and size of that window is saved instead.

While the system file uses the `.sdt` extension, the system window state file uses the `.sdt.state` extension.

If the Organizer finds a system window state file when opening a system file, the positions and sizes of the Organizer and editor windows are restored to the positions they had when the system file was saved. Note that the editor windows are not restored until they are opened from the Organizer.

Control Unit File

In addition to the System file (see "System File" on page 184), the Organizer also manages *Control Unit* files, `.scu` files. These control units are introduced to facilitate the workgroup (multiuser) support when working with an SDL system managed by Telelogic Tau, and merging the individual results to a common system file.

General

The multiuser support in Telelogic Tau is based on, from a revision control point of view, letting the user split the system file into several files. Information that is updated often should be split and stored in control unit files.

When and where to split the SDL structure is an active action taken by the user.

The multiuser mode is effective when control unit files are used. The use of the control unit files is optional, whereas the system file is mandatory. The Organizer thus always requires a system file and can manage multiple control unit files. By not taking advantage of control units simply means using the Organizer and system files as was done before, i.e. in previous versions of Telelogic Tau (3.0X/3.1X/3.2).

Definitions

A definition for a *system file* (`.sdt` file) from a revision control point of view: a file that contains structure and state information for a document system as seen by one Telelogic Tau user. This file is normally not considered as an essential part of the system and is not suitable for revision control as it is private to one user.

- The diagram structure information is common to all developers of the system in a multiuser environment.
- The state information is specific for one user.

A similar definition for a *control unit* file (`.scu` file): a file that contains structure information for a subset of a document system and which is common to all users that work with the document system. Control unit files are part of the document system and are suitable for revision control.

The control unit files can be inserted and made visible in the Organizer's file structure view. The control unit files contain information that is specific to the Organizer file structure for diagrams, modules and chapters.

Splitting the System File

The users control what parts of the file structure managed by the Organizer need a separate control unit file. When several users work on a system, the management of the system file may be difficult to synchronize. This problem is solved if every user or group have control over their own part of the system file.

The Organizer structure information is hence split into several files – control unit files. The decision of where and how many control unit files are needed is left to the users; the idea being to partition the diagram system according to the user's work responsibilities and assign control unit files to these partitions. An example can be a large SDL diagram system where there are several blocks on the system level developed in parallel by different developers. Each block could then be associated with a control unit file. Now, the blocks can be updated independently and the changes are shaping the local control unit files. There is no need to manually merge changes into the system file when the work is done – the management of control unit files performed by the Organizer also includes the merge.

The user decides how control unit files and the system file cooperate to produce the Organizer view of the document system. Two scenarios are possible here. In one scenario the user decides that the system file is valid and the diagram structure is read from it. This is done when the user opens the system file and loads the document system into the Organizer. In the other scenario the control unit files are used to load the Organizer view and automatically update it accordingly to a specific revision for the document system – the system file is used only to fill in document state information (if any is available). This is done with a menu command when control unit files are used to explicitly update the document system or some part of it.

An Example

To illustrate the use and contents of control unit files and facilitate the understanding, let us discuss the topic with a simple example as input.

Example 4 ––

Say we have the following system managed by the Organizer:

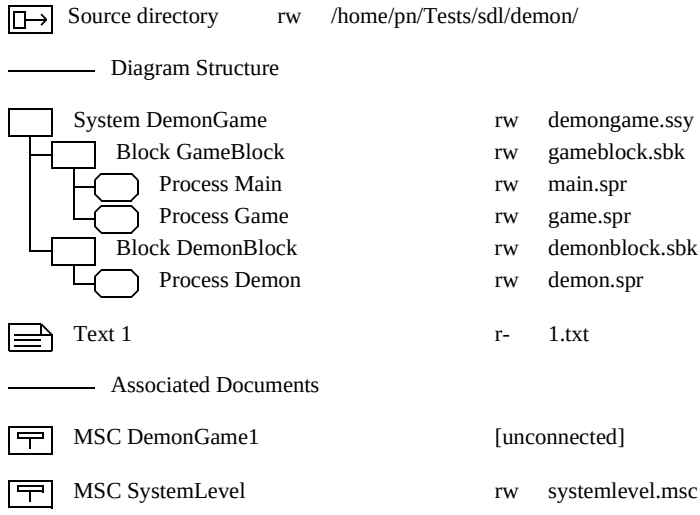

Figure 55: A simple system, managed by the Organizer

Let us say that we create control units (using the *Configuration > Group File* command) for the following items:

- One control unit that holds the entire system file together
- One control unit for the whole diagram structure
- One control unit for the system DemonGame
- One control unit for the block GameBlock.

We would now have a system that looks like this:

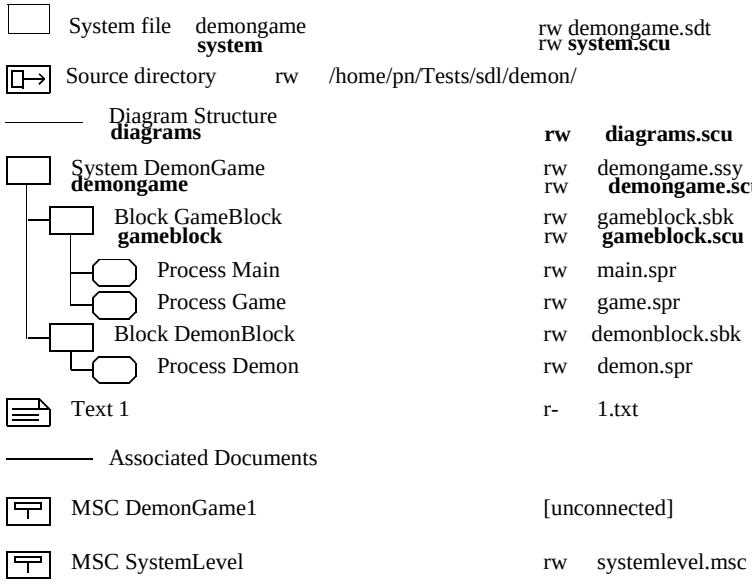

Figure 56: The same system, now with control units

The control units are depicted in bold face.

The [DIAGRAMS] section of the `demongame.sdt` file will contain:

```
[DIAGRAMS]
-2 102 system system.scu 0 0 - ""
-1 102 diagrams diagrams.scu 0 0 - ""
-1 23 "Diagram Structure" %unconnected% 0
0 102 demongame demongame.scu 0 0 - ""
0 1 DemonGame demongame.ssy 0 0 demongame ""
1 102 gameblock gameblock.scu 0 0 - ""
1 2 GameBlock gameblock.sbk 0 0 gameblock ""
2 5 Main main.spr 0 0 main ""
2 5 Game game.spr 0 0 game ""
1 2 DemonBlock demonblock.sbk 0 0 demonblock ""
2 5 Demon demon.spr 0 0 demon ""
0 33 1 1.txt 0 0 - ""
-1 23 "Associated Documents" %unconnected% 0
0 16 DemonGame1 %unconnected% 0 0 - ""
0 16 SystemLevel systemlevel.msc 0 0 - ""
```

The files `system.scu, diagrams.scu, demongame.scu` and `gameblock.scu` will contain the following information:

```
system.scu:
-1 102 diagrams diagrams.scu 0 0 - ""
-1 23 "Associated Documents" %unconnected% 0
0 16 DemonGame1 %unconnected% 0 0 - ""
0 16 SystemLevel systemlevel.msc 0 0 - ""

diagrams.scu:
-1 23 "Diagram Structure" %unconnected% 0
0 102 demongame demongame.scu 0 0 - ""
0 33 1 1.txt 0 0 - ""

demongame.scu:
0 1 DemonGame demongame.ssy 0 0 demongame ""
1 102 gameblock gameblock.scu 0 0 - ""
1 2 DemonBlock demonblock.sbk 0 0 demonblock ""
2 5 Demon demon.spr 0 0 demon ""

gameblock.scu:
0 2 GameBlock gameblock.sbk 0 0 gameblock ""
1 5 Main main.spr 0 0 main ""
1 5 Game game.spr 0 0 game ""
```

Format and Structure of the Control Unit File

Format

The format of the control unit file complies with the syntax of the [DIAGRAMS] section in the system file (see "Format of the System File" on page 185). There are however some exceptions:

The first line of the scu file indicates the source directory to use for relative files mentioned in the scu file. This information is saved in the same way as the same information is saved in the system file. By having this information in the scu file, it is possible to re-use/reference the same scu file from different system files with different source directories.

`<viewState>`
: This field is only used for the separation value - the value is therefore either 0 or 1.

`<timeStamp>`
: This field is not used - the value is always 0.

`<associations> ::= " <qualifiernameList> "`

A list of quoted qualifier names, each name denoting a document. See Example 5, below.

```
<dependencies> ::= " <qualifiernameList> "
```

A list of quoted qualifier names, each name denoting a document.

```
<qualifiernameList> ::= $ | (<qualifierName>
                          (, ($ | \NL) <qualifierName>)*)

<qualifierName> ::= ' <qualifierTuple>
                    (/<qualifierTuple>)* '

<qualifierTuple> ::= <integer> <string>
```

<string> is the name of the system file, chapter, module or diagram.

Example 5 ----------

```
" '23 Diagrams Area/1 DemonGame/2 DemonBlock', \
'23 Diagrams Area/1 DemonGame/2 GameBlock' "
```

Structure

Associations between Control Units and Document Structures

A control unit file is associated with a document structure in the Organizer. The control unit file holds information about the top node of the structure and information about lower level documents under the top node and/or other lower level control unit files.

Levels (Indents)

Document nodes in a control unit file have a relative level (i.e. `<indent>`) starting from 0. (Exceptions: a chapter node and an control unit file node associated with a chapter have always level -1). This gives the flexibility that the control unit file can be associated with a node on different levels in the system structure of documents. To get the actual level for a document node from an control unit, the control unit file node level should be added to the document's relative node level.

Example 6: For the control unit level 1 from Example 4 on page 197 –

```
0 2 GameBlock gameblock.sbk 0 0 gameblock ""
1 5 Main main.spr 0 0 main ""
1 5 Game game.spr 0 0 game ""
```

Hierarchy

The control unit files are hierarchical. The top control unit file is for the whole system of documents in the Organizer. We give this control unit file level (-2). If this file has an information line in the system file this means that the top of the whole system structure is found in that file. Other structure information about diagrams mentioned in the system file is then only a mirror of the structure information found in the control unit hierarchy.

Example 7:

```
-2 102 system system.scu 0 0 - ""
```

Modularity

The user can choose to use a control unit file only for a specific chapter, module or diagram structure (without a top level control unit system file). This means that not all information in the system file is a mirror for structure information from control unit files. Some of the information can be actual structure information that is not controlled by control unit files. When the Organizer expands structure information from control unit files it tries to match that information against the contents in the system file. This matching is necessary because only the system file holds state information about documents. The structure information from control unit files takes precedence over structure information found in the system file. If there is an inconsistency between the control unit file structure information and the system file, the control unit information takes precedence and the SDL suite contents are treated as an error situation. Say, for instance, that according to a control unit file there are three blocks in an SDL system but there are four blocks in the system file. The extra block is probably old information.

Example 8: A control unit file for a chapter

```
-1 102 diagrams diagrams.scu 0 0 - ""
```

Order of Appearance

A control unit information line in the system file is followed by a mirror information line of the top node that the control unit file is associated with. If the Organizer cannot open the control unit hierarchy the mirror information will function as a back-up document structure information.

Example 9 ----------

```
-1 102 diagrams diagrams.scu 0 0 - ""
-1 23 "Diagram Structure" %unconnected% 0
```

Batch Facilities

the SDL suite supports operations in batch mode, such as *printing* the documentation, *analyzing* an SDL structure and *generating code*. These operations are managed by the Organizer in a batch, windowless mode.

Batch Syntax

The syntax for a batch command is:

```
sdtbatch <systemfile> <options>
```

Where `<systemfile>` and `<options>` are described in "Batch Options" on page 207.

When the `sdtbatch` command is issued, the Organizer interprets the command line arguments, performs the requested operation and then terminates. Error logging is performed by the Organizer.

Information concerning the batch job is directed the standard output device.

> **Note:**
>
> When executing a Batch command, external service management via the PostMaster interface is disabled.

Print Multiple Files

Syntax:

```
sdtbatch -p[rint] systemfile \
         [-s[elect] documentfile] \
         [-o[ption] optionfile]
```

Prints the contents in the system file. Corresponds to clicking the *Print* quick button.

Print One File

Syntax:

```
sdtbatch -p[rint] \
         -s[elect] documentfile \
         [-o[ption] optionfile]
```

Prints one file.

Compare Two SDL Diagram Files

Note:

This sdtbatch command produces a textual difference report. To get a graphical difference report, consider using the -grdiff option with the sdt startup command. Read more about this in chapter 2, *Introduction to the SDL Suite*.

Syntax:

```
sdtbatch -d[iff] [ -b ] [ -n ] \
          documentfile1 documentfile2
```

Compares two SDL diagram files. By default, a summary of how many differences that were found is printed. When using the `-b` option (verbose), information about every differing object is presented, including an SDT reference to the object in question. The use of the `-n` option corresponds to setting the option Ignore moved or resized objects to off. Normally this means that moved and resized symbols will not be detected as being different.

Analyze

Syntax:

```
sdtbatch -a[nalyse] systemfile \
        [-s[elect] diagramfile] \
        [-o[ption] optionfile]
```

Analyzes the contents in the system file. Corresponds to the *Analyze* quick button.

Make

Syntax:

```
sdtbatch -m[ake] systemfile \
         [-s[elect] diagramfile] \
         [-o[ption] optionfile]
```

Makes a target as specified by the system file. Corresponds to the *Make* quick button.

Extract Text

Syntax:

```
sdtbatch -extract [ -comment ] \
```

```
[ -keyword <keyword> ] <diagram file>
```

Extracts text from the diagram file. As default, all text in the diagram is shown. By using the *-comment* option, only text in comment symbols and /* C-style comments */ in other symbols is shown. The *-keyword* option limits the shown text to text from symbols containing the <keyword>. The <keyword> itself is not shown. For instance, if you have a comment symbol with the following text:

```
diagramversion 1.44
```

Then...

```
sdtbatch -extract -comment \
         -keyword diagramversion a.ssy
```

...will produce the following output:

```
1.44
```

Show License Information

Syntax:

```
sdtbatch -licenseinfo
```

Show information about license holders and number of available licenses, as textual output in the console window where the command was entered.

List Files

Syntax:

```
sdtbatch -files systemfile
```

Lists all files referenced from the system file, as textual output in the console window where the command was entered.

Pack Files

Syntax:

```
sdtbatch -pack systemfile [ archivefile ]
```

Create an archive file (*.tgz) for a system, using default pack settings. If the archive file name is not given, then an archive file will be created in target directory, with a default name.

Change Case

Syntax:

```
sdtbatch -changecase cross-reference-file
```

Change the case for all words found in all referenced files in the cross reference file to the case used in the declaration.

The command will also change the case for all keywords. These should be located in a special keyword file found in the same directory, with the same name as the name of the given cross reference file but with the extension `.key`.

The command does only work with the standard SDL text editor.

To generate a more extensive cross reference file and a file containing keywords, run the `sdtbatch -analyze` command with the options

```
SDLKeywordFile=True

SetPredefinedXRef=True
```

added to the option file. The analyzer will then generate more information in the cross reference file, case-sensitive `.xrf`, and furthermore a keyword file, case-sensitive.key, in the target directory of the system.

Configuration Update

Syntax:

```
sdtbatch -r[ecursive] systemfile \
         [-s[elect] diagramfile] [-u[pdate]]
```

Does a recursive update of the structure in the Organizer according to the Control Unit files in the system file. Corresponds to the *Configuration > Update* menu choice having the diagram file selected. Without the -s option the action is as if the systemfile has been selected.

Checking diagrams for duplicated object IDs

Syntax:

```
sdtbatch -checkdiagrams systemfile
sdtbatch -checkandfixdiagrams systemfile
```

Normally each object in an SDL diagram have a unique ID within the diagram. This ID can be shown for example when using the SDL editor command Show GR Reference. For some existing SDL diagrams, espe-

cially diagrams that have been created using the Merge or Add Differences functions, the same ID has been used by more than one object. This will result in unpredictable behaviour when using GR References, e.g. when transforming the system to case-sensitive SDL or when an analysis error should be shown, the wrong symbol might appear. Also if the two objects having the same ID are on the same SDL Page, the lines can be connected to symbols in a way that is different from the way you expect by looking at the geometrical layout of the lines. This can give strange errors when generating SDL/PR, giving strange analysis errors. This problem has been fixed in versions 4.2.7, 4.3.3, and 4.4, therefore diagrams that are saved with these or later versions will not have this problem.

To test the diagrams for duplicated IDs you can run the batch command `-checkdiagrams`. The command will produce a report on duplicates for all the SDL diagrams.

Whenever there are problems with duplicates you can run `-checkandfixdiagrams`. This will fix all duplicates and save the diagrams such that all diagrams are free from duplicated IDs.

Batch Options

- `<systemfile>`

  `<systemfile>` is an Organizer *System File*. A valid system file should be supplied.

- `-select <file>`

  `<file>`, which should be included in the document structure, (associated documents are also included if the `-print` option specified) becomes the selected object upon which the command is applied. This option corresponds to a manual selection of the document in the Organizer drawing area.

  A full file path for the file must be supplied.

- `-o[ption] <optionfile>`

  `<optionfile>` is a file with the same syntax as a *System File*, containing options which supersede the options found in the system file. An optional section CPP2SDLOPTIONS can manually be entered into the option file to supersede the SDL systems cpp2sdl import specification files. An arbitrary number of options may be sup-

plied, but options in the file must be preceded with an option section field as found in the system file.

Note:

Note that if the options contains a filename in quotes all backslashes must be escaped with one more backslash. Example of correct filename uses:

```
PrinterFile=d:\tmp\printfile.ps
```

or

```
PrinterFile="d:\\tmp\\printfile.ps"
```

- `-u[pdate]`

  If this option is supplied, the system file will be saved after the batch operation has finished.

Windows and UNIX File Compatibility

Systems can be created with Telelogic Tau on both the UNIX and Windows platforms. To allow systems to be accessed on both platforms, Telelogic Tau provides a file compatibility concept.

The *drive table* contains a mapping between Windows and UNIX directory paths. This mapping is included in the system file to allow file specifications in the system file to be translated to the correct format on both platforms. See "The Drives Section" on page 187 for more information.

The menu choice *PC Drives* is available in the Organizer's *File* menu, and allows editing the drive table that is stored in the system file. See "PC Drives" on page 72 for more information.

The drive table takes care of translating absolute paths between the two different file name formats (UNIX and Windows). On a UNIX system, some restrictions apply to file names to be compatible with Windows. These restrictions are checked whenever the user changes a file specification that is stored in the system file; if they are not followed, an error dialog is shown and the user is returned to the dialog where the file was specified. See "Filename Error Dialogs (UNIX only)" on page 33 for more information.

File specifications are always stored in the correct format according to the current platform.

In general, diagrams with file specifications that are of incorrect format or that cannot be translated to the correct format, are marked as Invalid in the Organizer.