

**Introduction to
Telelogic Tau
SDL Suite**

200109





Copyright by Telelogic AB 1991-2001

This publication contains course materials and examples of applications of the Program and is owned by Telelogic and is protected by national copyright laws and international copyright treaties. Telelogic grants you the right to use the training course material and the enclosed examples on one computer or in one local computer network at any one time. This does not convey to you an interest in or to the publication, but only a limited right of use.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form without the written permission of Telelogic.

Telelogic does not warrant that this training course material and the enclosed examples will meet your requirements or that the operation of the Program will be according to the information in this publication, uninterrupted and error free. You are solely responsible for the selection and operation of the Program to achieve your intended results and for the results actually obtained.

Information in this publication is subject to change without notice.

Introduction to Telelogic Tau SDL Suite

Introduction.....	1
Starting up Telelogic Tau SDL Suite.....	2
The Organizer	2
The Preference Manager	5
Help Viewer	6
The SDL Editor	7
Block Diagrams	8
Process Diagrams.....	10
Editing Text	12
On-line Syntax Check.....	12
Grammar Help	13
The Analyzer	14
Trace static errors in an SDL System	15
The Index Viewer	16
The MSC Editor	17
The Simulator	19
The Coverage Viewer	23
The Validator	24

Introduction

Telelogic Tau SDL Suite is a tool for designing and implementing real-time software.

The SDL Suite consists of a graphical editor and syntax analyzer, a simulator and validator and several optimized code generators for compilation to executable code. The SDL Suite consists of the tools shown in *Figure 1*.

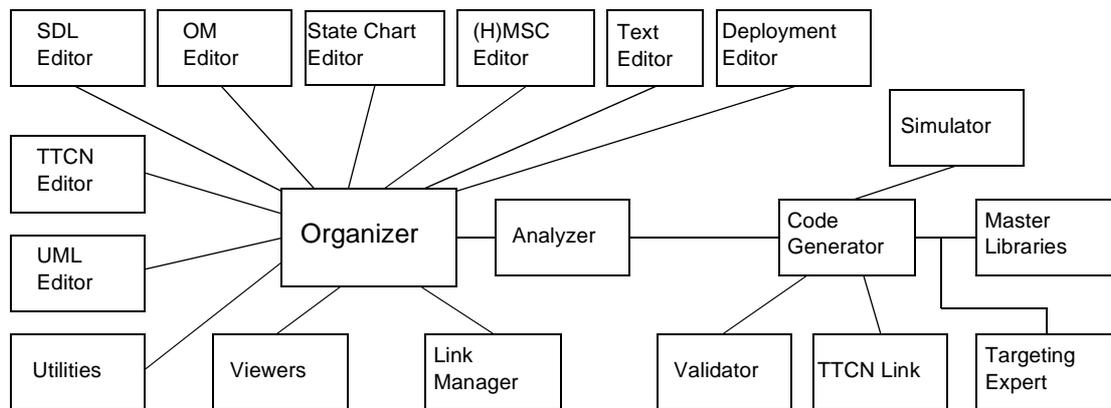


Figure 1 Telelogic Tau SDL Suite

Telelogic Tau SDL Suite is available on UNIX and Windows platforms.

This document describes the basic functions of the Telelogic Tau SDL Suite.

Starting up Telelogic Tau SDL Suite

UNIX `>sdt`
or
`>tau`

Windows Use the **Start/Program files** and select the Telelogic Tau program.

The SDL Suite starts and the Organizer window appears.

The Organizer

The Organizer is the tool that assists you when working with the SDL and MSC diagrams. The Organizer can handle text files, object models and c-files (see *Figure 2*). In fact, any file can be referenced in the Organizer. It is also the Organizer that is managing the other tools in the SDL Suite.

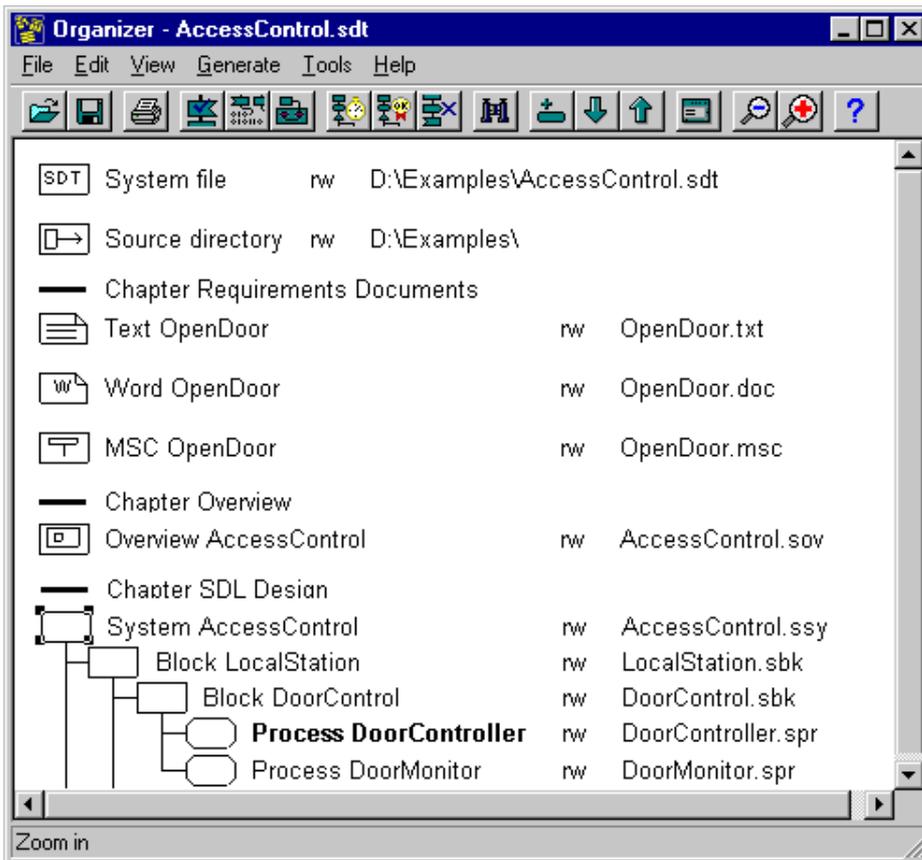


Figure 2 Organizer window

A general introduction to some important commands within the Organizer are found in *Table 1*. Note that several of these commands are also available in the context-sensitive pop-up menu (right mouse button).

Table 1: Frequent Organizer Menu Commands

Menu / Command	Action
File / New...	Creates a new system file.
File / Open...	Opens an old system file.
File / Save...	Saves the current settings in the system file.
File / Print...	Prints the whole structure or parts of the structure.
File / Compare System...	Compares the system in the Organizer with the system in a specified <code>.sdt</code> file, with the possibility to add or remove diagrams that differ.
Edit / Add New...	Adds a new diagram/module/chapter to the contents of the Organizer.
Edit / Add Existing...	Adds a file to the contents of the Organizer.
View / View Options...	Sets the view of the Organizer window.
Generate / Analyze...	Starts analyzing the selected system.
Generate / Make...	Starts the make process, generates code and an executable Simulator / Validator / Application /...
Generate / SDL Overview...	Starts to generate an SDL Overview diagram (a transparent SDL view) for the selected diagram in the Organizer.
Generate / Convert to GR...	Converts an SDL PR file to SDL GR diagrams.
Tools / Search...	Starts the search and replace tool.
Tools / SDL / Type Viewer	Starts the Type Viewer, used to look at the inheritance trees and instantiation information.
Tools / SDL / Coverage Viewer	Starts the Coverage Viewer, to look at coverage information from simulations and validations.
Tools / SDL / Index Viewer	Starts the Index Viewer, to look at SDL definitions and where the definitions are used.
Tools / SDL / Simulator UI	Starts the Simulator user interface.
Tools / SDL / Validator UI	Starts the Validator user interface.
Tools / Preference Manager	Starts the preference manager, where the interface and behavior of the other tools can be customized.

The Organizer has fast access to the most frequent commands by:

- a tool bar with quick buttons (see *Figure 3* and *Table 2*)
- a context sensitive pop-up menu (on the right mouse button)
- keyboard short-cuts.

The following quick buttons are available in the Organizer:



Figure 3 The Organizer Tool Bar

Some quick buttons correspond to menu commands, while other quick buttons perform a sequence of actions.

Table 2: Useful Quick Button Commands

Command	Action
Simulate	Analyses the selected SDL system, makes a Simulator and opens the Simulator in a Simulator UI.
Validate	Analyses the selected SDL system, makes a Validator and opens the Validator in a Validator UI.
Generate Index	Analyses the selected system, generates a cross reference file and opens that file in the Index Viewer.
Move Down	Move the selected diagram down.
Move Up	Move the selected diagram up.

The Preference Manager

The Preference Manager allows you to customize the appearance and behavior of the SDL Suite. To start the Preference Manager, choose the command **Preference Manager** from the Organizer **Tools** menu and the Preference Manager window appears, see *Figure 4*.

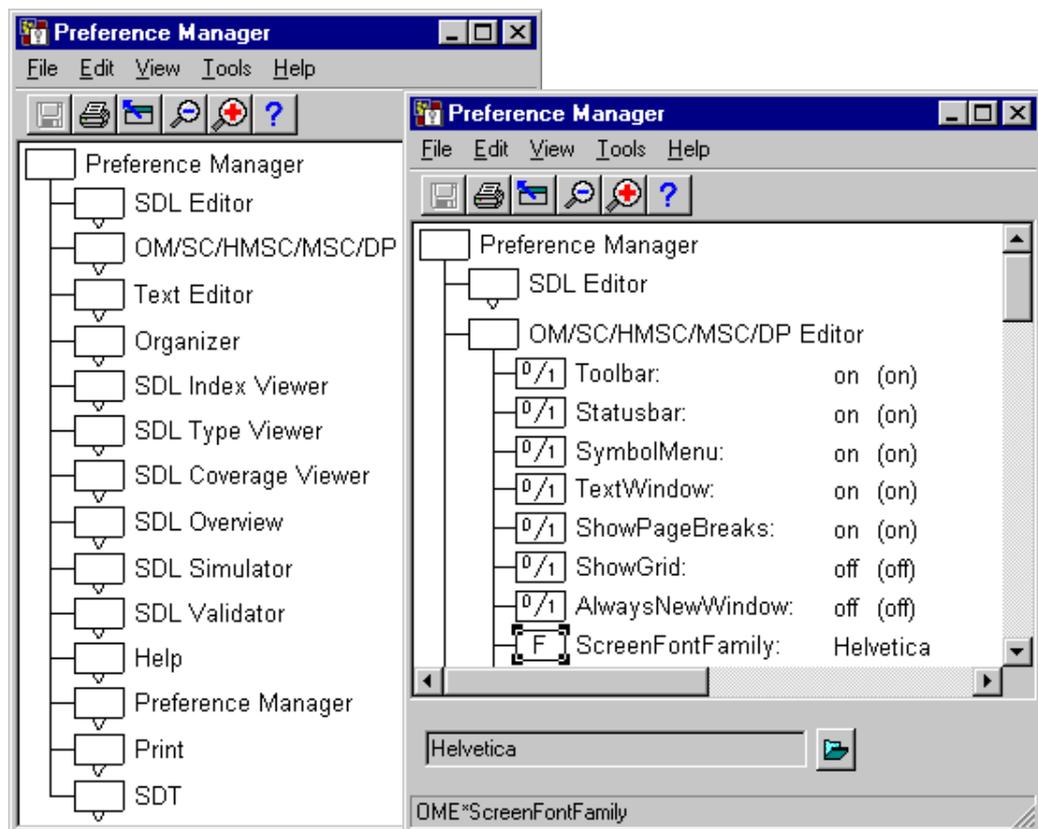


Figure 4 Preference Manager windows

To show the preference values you can:

- double-click on an icon
- select the icon and choose the **Expand** command under the **View** menu
- press the right mouse button on the icon and choose the **Expand** command in the popup menu

Perform the **Save** and then the **Exit** command from the **File** menu in the Preferences window. Your settings have now been saved and will be read by the other SDL Suite tools when they start up.

Help Viewer

A Help menu is available in most main and sub windows of the SDL Suite. However, the menu choices in this menu are not the same in each tool or on each platform, but they work in the same way: When you select a menu choice in the Help menu, a help viewer will be opened with the corresponding help topic, see *Figure 5*.

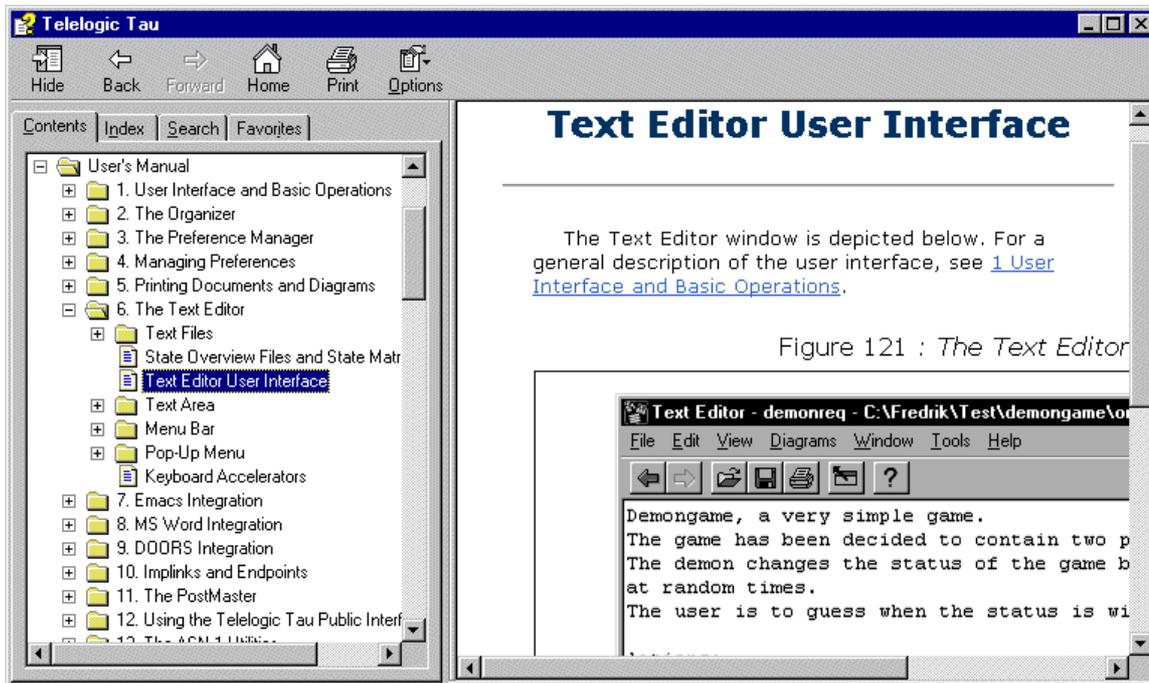


Figure 5 Help Viewer window

In the help viewer you can perform the actions described in *Table 3*.

Table 3: Useful Help Viewer Commands

Command	Action
Index	Opens the help viewer with an index of all entries in the Telelogic Tau documentation.
Search	Starts a textual search across all help files.

The SDL Editor

The SDL Editor is the tool that lets you create, view and edit your SDL diagrams. The editor has different modes, depending on the kind of edited diagram, for example block or process diagram. The modes differ with respect to on-line syntax and the symbols in the symbol menu. A few commands are also affected by the mode.

The most frequently used commands in the SDL Editor are viewed in *Table 4*. Note that several of these commands are also available in the context-sensitive pop-up menu (right mouse button).

Table 4: Frequent SDL Editor Commands

Menu / Command	Action
File / Save	Save the current diagram.
File / Print...	Print the current diagram.
Edit / Cut, Copy, Paste	Graphical clipboard features.
Edit / Drawing Size...	Set the size of the diagram drawing area.
View / Window Options..., Diagram Options..., Editor Options...	Set some options for the Window, Diagram or Editor.
Pages / Edit...	Editing features for diagram pages: Add pages, Rename pages, Edit pages, Copy pages,...
Diagrams / ...	Switch between the different diagram buffers in the editor.
Window / New Window	Open a new SDL Editor window.
Window / Grammar Help	Open the Grammar Help window.
Window / Signal Dictionary	Open the Signal Dictionary window.

The quick buttons available in the SDL Editor are shown in *Figure 6*.

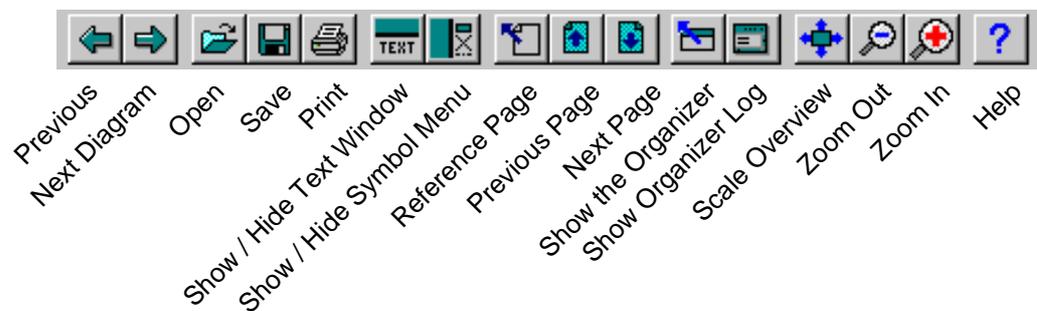


Figure 6 The SDL Editor Tool Bar

Table 5: Useful Quick Button Commands

Command	Action
Show / Hide Text Window	Toggle the text window.
Show / Hide Symbol Menu	Toggle the symbol menu.
Reference Page	Edit the diagram where this diagram is referenced.
Previous / Next Page	Show the previous / next page.
Pop-up the Organizer	Show the Organizer.
Scale Overview	Set the scale to fit the diagram into the size of the window.

Block Diagrams

The appearance of the SDL Editor when block diagrams are created and edited is shown in *Figure 7*.

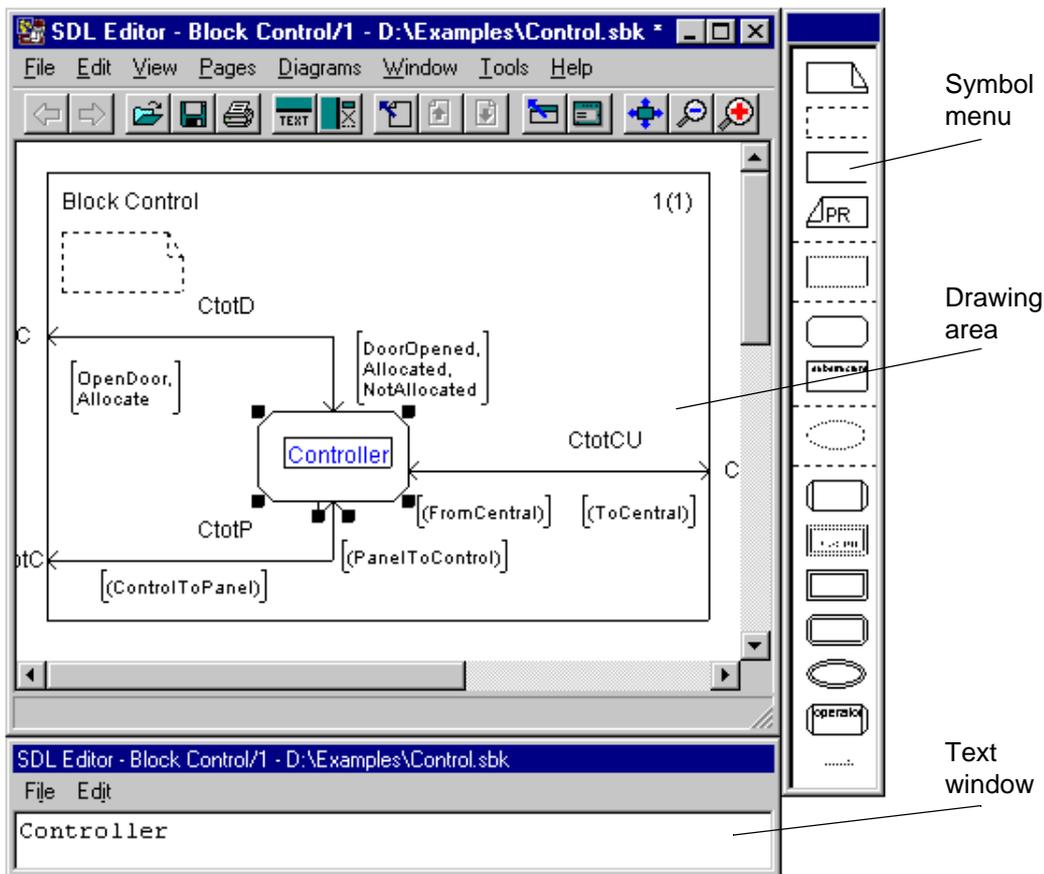


Figure 7 SDL Editor window in block diagram mode

Draw symbols

Symbols can be placed either by double-clicking in the symbol menu or manually by selecting and dragging from the symbol menu. When selecting a symbol in the symbol menu, the cursor is shaped as an arrow.

A 4-Arrow cursor will be shown when a click + drag will result in moving the object.

To resize a symbol, grab one of the corners of the symbol, and drag.

The text symbol is shrinkable to save space on a drawing. If you double-click in a text symbol it will assume the smallest size, one more double-click will restore the size again. On a print-out, the text inside a shrunk text symbol is fully expanded on a separate page.

Symbols can be copied and pasted between different editor modes provided that the copied information is allowed in that editor mode.

Draw channels and signal routes

Channels can be inserted if you select a block symbol, grab the handle on the bottom of the symbol and drag.

Signal routes can be inserted if you select a process symbol, grab the left handle and drag. Observe that there are two handles on the process symbol. The second handle will produce a Create Line (dashed) which is used for indicating a Parent-Offspring relationship, see *Figure 8*.

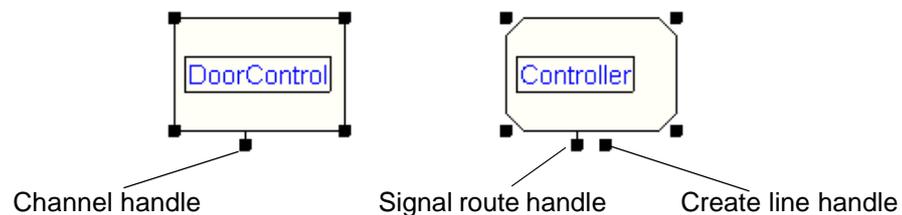


Figure 8 Channel and Signal Route handles

The components of a Channel (identical to Signal Route) between a block and the environment are shown in *Figure 9*.

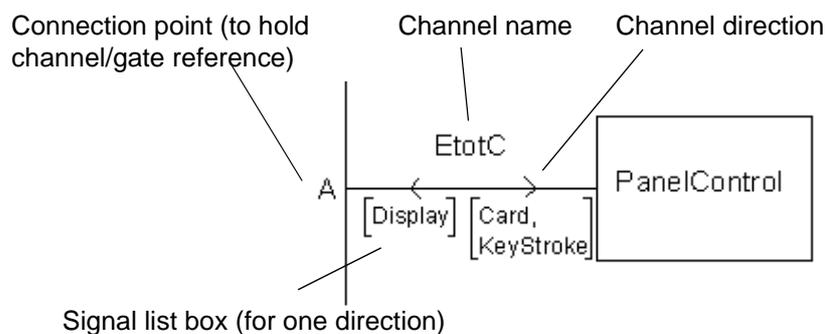


Figure 9 Channel components

The placement of channels and signal routes can be interrupted by either pressing the <ESC> key or double-clicking on the left mouse button.

The channel or signal route direction can be changed with the menu commands redirect, bidirect and unidirect, see *Table 6*.

Table 6: Channel and signal route menu commands

Command	Action
Edit / Redirect	Redirect a channel or signal route.
Edit / Bidirect	Bidirect a channel or signal route.
Edit / Unidirect	Unidirect a bidirectional channel or signal route.

Process Diagrams

The appearance of the SDL Editor when process diagrams are created and edited is shown in *Figure 10*.

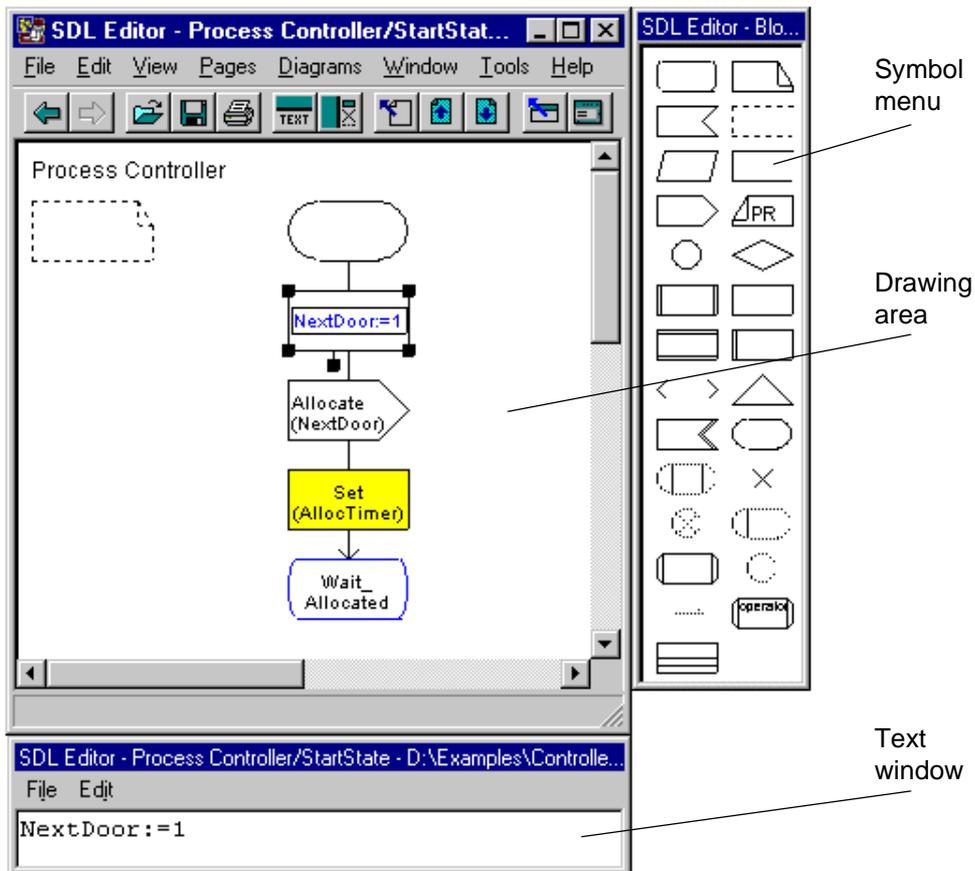


Figure 10 SDL Editor window in process diagram mode

Apart from the commands described in *Table 4*, there are some more useful commands when editing process diagrams. Note that several of these commands are also available in the context-sensitive pop-up menu (right mouse button).

Table 7: Frequent SDL Editor Commands for Process Diagrams

Menu / Command	Action
Edit / Insert Paste	Pastes the contents of the clipboard into a connected flow, trying to preserve the flow lines.
Edit / Flip	Mirrors the graphical appearance of certain flow symbols vertically.
Edit / Select Tail	Selects all items following the selected item. Also available with the keyboard short-cut <ctrl> - t.
Tools / Tidy Up...	Redraw your diagram with auto layout of symbols.
Tools / Connect to Text Editor	An external Text Editor can be used to edit large texts in symbols.

Add symbols

When double-click is used, the new symbol is automatically connected to the symbol selected in the flow. The newly inserted symbol is now selected, so that you can directly double-click on the next symbol to be inserted in the flow. The great benefit is that the flow lines do not have to be manually connected. This is called Auto placement.

The selection is not moved down when double-clicking, if <SHIFT> is pressed. This can be useful if you for example want to place several INPUT symbols connected to the same STATE symbol.

Connecting flowlines

Flowlines can be inserted if you select a symbol, grab the handle on the bottom of the symbol and drag.

Flowlines can also be inserted automatically. If you use the auto placement for symbols the symbols will automatically be connected with flowlines provided that it is a syntactically correct flow.

The placement of flowlines can be interrupted either by pressing the <ESC> key or double-clicking on the left mouse button.

If you want to insert a symbol into an existing flow, just place the symbol from the symbol menu on the flowline where you want to insert it or select the flowline (or the symbol above) and double click in the symbol menu on the symbol you want to insert.

Multiple selection

A group of elements or an area can be selected by pressing and dragging the left mouse button to create a box that completely surrounds the group or area.

Several symbols can be selected after each other by first selecting one of the symbols, pressing <shift> and then selecting the symbols one by one. From a group of selected symbols it is possible to deselect one by one in a similar way.

There is also a possibility to select one flow symbol and then automatically select all symbols in the branches from that symbol. This is done automatically with the Select Tail command (see *Table 7*), which is also available as the keyboard short-cut <ctrl>- t.

Editing Text

If you have a symbol selected, you can write text into the symbol. You can place a text cursor anywhere in a text inside a symbol, see *Figure 11*.

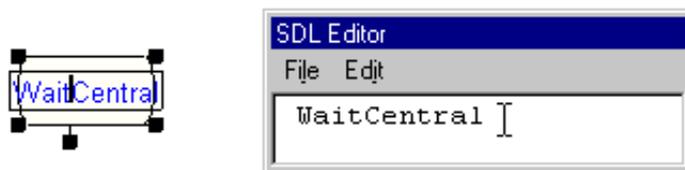


Figure 11 Text selected in symbol is visible in text window

If the symbol is unselected an I-beam cursor will be shown and a click will result in entering text editing mode.

Observe that in the case where a selected object has more than one text area (e.g. a channel has name, signal list and connection point) it is the name of the object that the text will be entered into. Each text area can be selected individually, though.

Text handling, like clipboard operations, are handled in the text window. When the cursor is in the text window it is constantly shaped as an I-beam.

On-line Syntax Check

The SDL Editor checks that the symbols you add to a diagram are in accordance with the syntactic rules imposed by SDL.

Some texts in the editor are subject to syntactic checks as soon as they are changed. Errors detected during syntax checks will be displayed in the textual object by a red underlining.

Grammar Help

Grammar Help is a very convenient tool when you define a text-based definition and you need help with the exact syntax. Grammar Help is useful when defining for example newtypes, syntypes, and signals, see *Figure 12*.

The Grammar Help is started in the **SDL Editor** under the **Window** menu with the command **Grammar Help**.

Double-click here to insert a template into the text symbol.

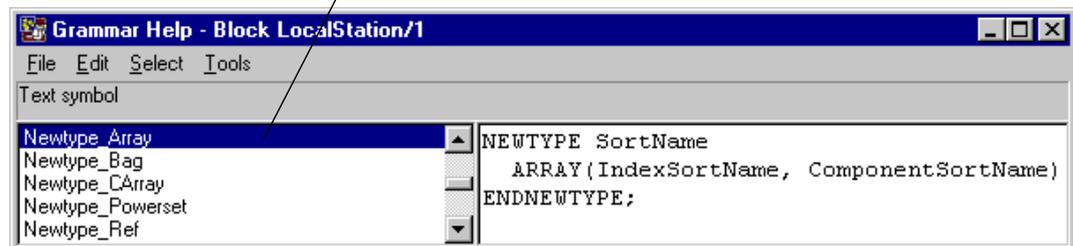


Figure 12 Grammar help window

The Grammar Help is context sensitive: it shows relevant information for the selected symbol.

The sections that starts with GRAMMAR in the Grammar Help describes the BNF (Backus-Naur Form) production rules for different concepts available in the selected symbol. Chapter references to ITU-T recommendation Z.100 are also available for reading about the definitions of the different concepts.

The templates are perhaps the most useful part of the Grammar Help functionality. The templates are items that do **not** start with GRAMMAR. Predefined templates for certain design constructs are available. Identifier names are generic (i.e. SortName, ComponentName etc.) and should be changed to relevant names.

The templates can be automatically inserted into the selected symbol by double-clicking on the template name in the left Grammar Help window. The insertion can also be cancelled by the **Undo** command in the **Edit** menu of the **Grammar Help** window.

The Analyzer

The SDL Analyzer (see *Figure 13*) checks that SDL diagrams are designed according to syntactic and semantic rules in Z.100.

An analysis is started by selecting the part you want to analyse in the Organizer and then do one of the following:

- Choose the **Analyze...** command under the **Generate** menu in the **Organizer**. Set the options for the analysis (semantic analysis can, for example, be omitted) in the dialogue that appears and then press the **Analyze** or **Full Analyze** button.
- Press the **Analyze** quick button in the Organizer. In this case, the latest specified options in the Analyze dialogue will be used.

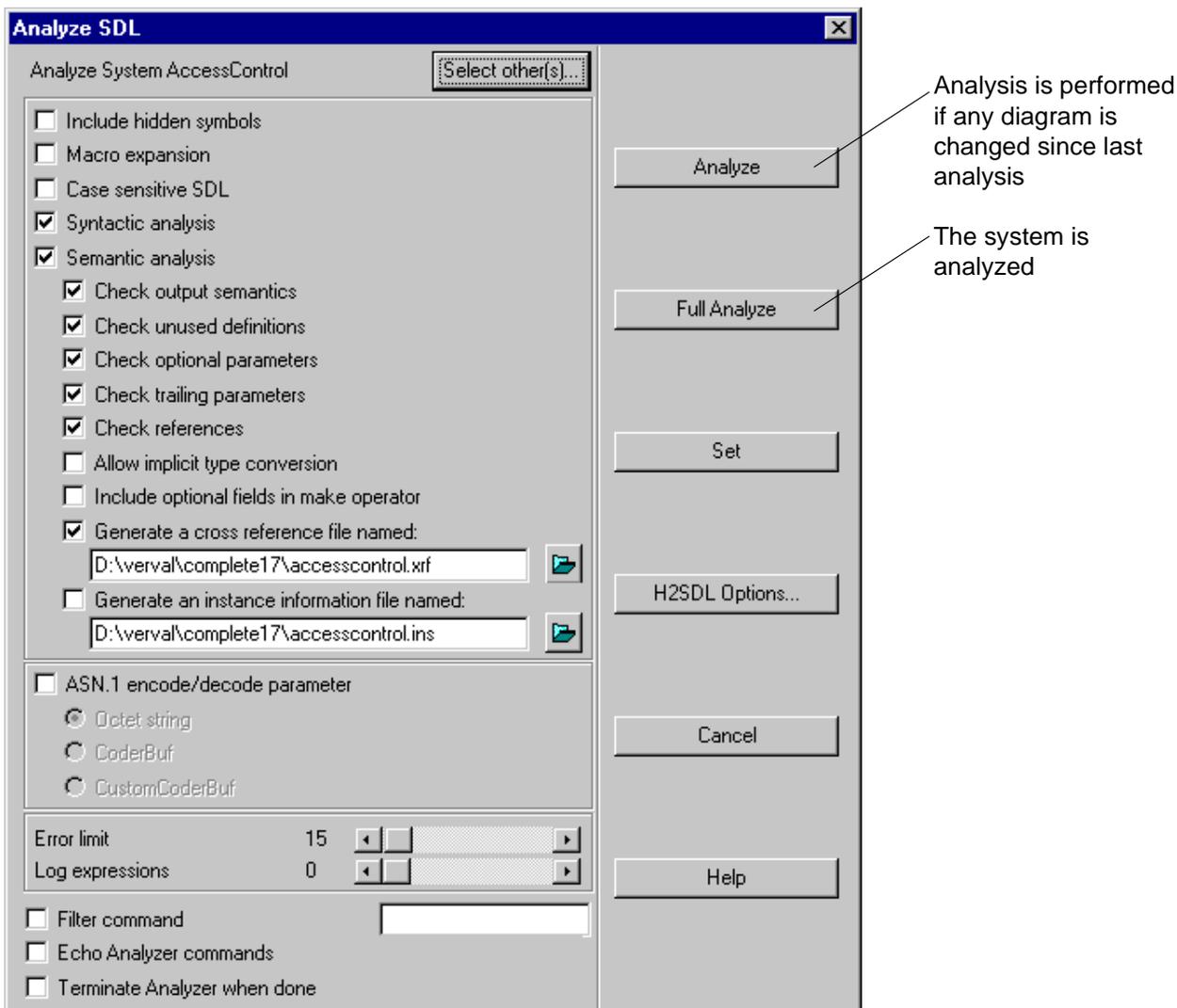


Figure 13 Analyzer window

The results of the Analyzer are appended to the Organizer Log Window.

Trace static errors in an SDL System

If any errors or warnings are found during the analysis the Organizer Log window will appear, and inform you about what happens during the analysis. All errors found during the different phases of the analyzer will be displayed in this window, see *Figure 14*.

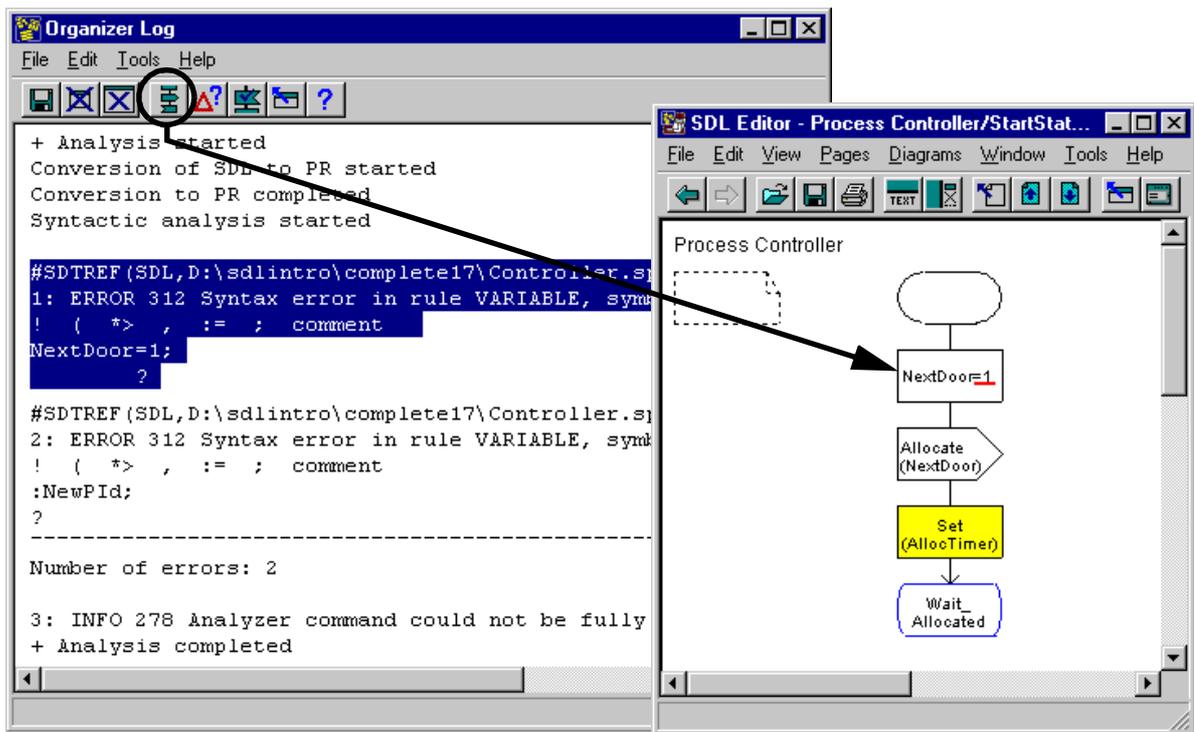


Figure 14 Show error in Organizer Log window

The SDL suite provides a nice feature for displaying the source of an analysis error:

1. Select the error (or warning) message in the Organizer log window by dragging the mouse.
2. Select the menu choice **Show Error** from the Tools menu or use the **Show Error** quick button in the Organizer log window 
3. The symbol where the error has been detected is displayed in an SDL Editor window.

A repeated analysis (with previous settings) is conveniently done in the Organizer Log by the **Analyze** quick button. 

If you select an error in the Organizer Log window you can use the **Help on error** quick button to get a brief help message about the error. 

The Index Viewer

The Index Viewer shows all the user defined identifiers in an SDL system in a graphical way. The information is generated at the end of the semantic analysis. The Index Viewer (see *Figure 15*) is a useful feature for finding SDL information in a large system, since it keeps track of both the definition of an entity and all references to that entity.

The Index Viewer is started in the **Tools** menu in the Organizer under the sub-menu **SDL**, using the command **Index Viewer**. It can also be started by using the **Generate Index** quick button in the Organizer.

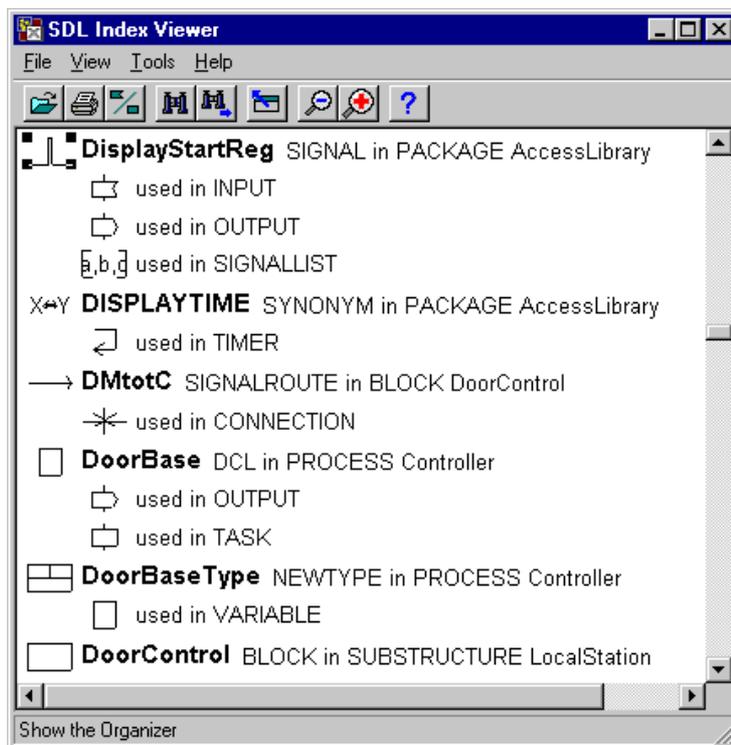


Figure 15 Index viewer window

The Index Viewer appears with information about all identifiers in the SDL System. Both the definitions and the references of an identifier can be found. The predefined data types of SDL are also shown.

The sort order can be changed with the **Toggle Order**  quick button. The information is now presented with sort order **Type and name** instead of **Name**. Press the quick button again to have the information sorted by **Name**. Start typing the name in the Viewer window to search for an identifier.

Select the **Tools/Show Definition** command (in the References window) to show the definition in the SDL Editor.

The MSC Editor

The MSC Editor is the tool that lets you create, view and edit your MSC diagrams according to the ITU-T recommendation Z.120. Here you can design your system scenarios/use cases and see trace outputs from simulations and/or validations, see *Figure 16*.

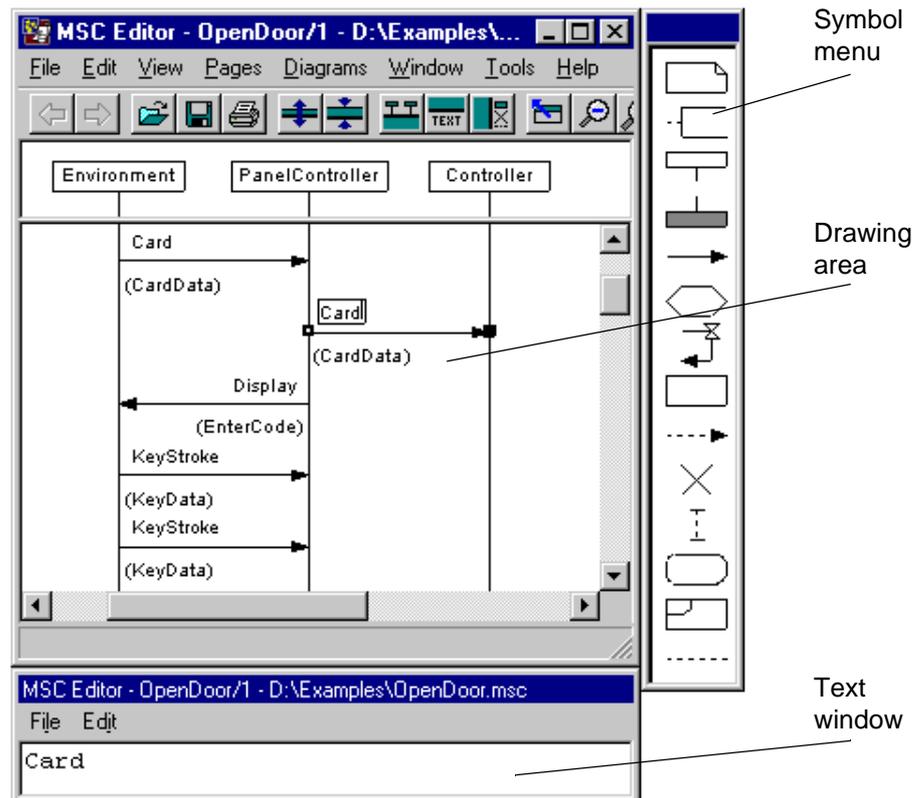


Figure 16 MSC Editor window

The Clipboard functionality could be used for copying similar messages, for example.

The diagram size can be changed by the **Drawing Size** command in the **Edit** menu.

The most frequently used commands in the MSC Editor are viewed in *Table 8*. Note that several of these commands are also available in the context-sensitive pop-up menu (right mouse button)

Table 8: Frequent MSC Editor Commands

Menu / Command	Action
File / Save	Save the current diagram.
Edit / Cut, Copy, Paste	Graphical clipboard features.
Edit / Drawing Size...	Set the size of the diagram drawing area.
Diagrams / ...	Switch between the different diagram buffers in the editor.
Window / New Window	Open a new MSC Editor window.
Window / Info Window	Open the Info window, displaying information about the selected symbol.

The quick buttons in the MSC Editor are shown in *Figure 17*.



Figure 17 The MSC Editor Tool Bar

Table 9: Useful Quick Button Commands

Command	Action
Make Space	Insert space after the selected event.
Remove Space	Remove space after the selected event.
Show / Hide Instance Ruler	Toggle the instance ruler.
Show / Hide Text Window	Toggle the text window.
Toggle Symbol Menu	Show / hide the symbol menu.
Pop-up the Organizer	Show the Organizer.

The Simulator

The Simulator is used to test and learn the dynamic behavior of the system specification. The Simulator works as a debugger on SDL level. During simulations it is possible to have MSC, SDL and textual traces. To facilitate simulation commands can be logged and later included as scripts.

If you want to create a simulator, you can use one of the following ways:

- Select the system in the Organizer. Choose the **Make...** command in the **Generate** menu. Set the **Standard kernel** option to one of the **Simulation** options in the dialog which appears. Press the **Set** button. Press the **Simulate** quick button in the Organizer.
- Choose the system you want to create a simulator for in the Organizer and then choose the **Make...** command under the **Generate** menu in the Organizer. Set the **Standard kernel** option to **Simulation** in the dialog which appears and then press the **Make** button. When you have created a simulator you can start the simulation by choosing the **Simulator UI** command in the **SDL** sub-menu in the **Tools** menu in the Organizer.

The Simulator UI window appears, see *Figure 18*.

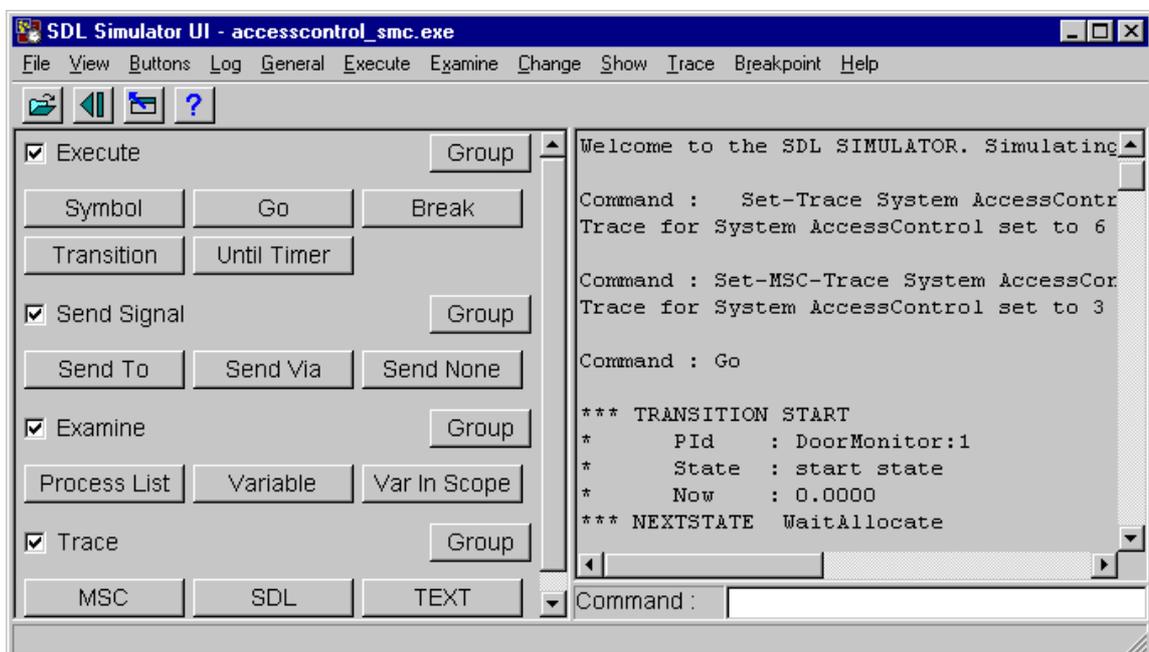


Figure 18 The Simulator window

Table 10: Useful Simulator Button /Menu Commands

Group / Button Menu / Choice	Textual command	Action
Send Signal/ Send To	output-to	Sends a signal from the environment to an SDL process instance. Expected parameters: signal_name [parameter_val_list] proc_name
Send Signal/ Send Via	output-via	Sends a signal from the environment into the system via a certain channel. Expected parameters: signal_name [parameter_val_list] channel
Examine / Process List	list-process	Lists all process instances associated with the specified process type. Parameters: <Optional process name>
General / Start SDL Env	start-sdl-env	Starts handling of the SDL environment (connects to other applications via a broadcast socket mechanism).
Execute / Go	go	Will execute the system until there are no more transitions that can execute.
Execute / Transition	next-transition	Executes one transition in the system according to the ready queue.
Execute / Into Stmt	step-statement	Executes one statement in the system (the finest granularity). Procedure calls are expanded and each assignment is treated as one statement.
Execute / Until Time	proceed-until	The execution of the simulation is resumed and is active when the value of the simulation time is equal to the time value given as parameter. Parameters: <Time value>
Execute / Command Script...	include-file	Reads a simulator script from a specified text file. Expected parameters: file_name
Execute / Break	<return>	Breaks the execution and returns to the monitor.
View / Set Scope	set-scope	Sets the scope. Expected parameters: proc_name
View / Ready Q	list-ready-queue	Shows the order in which the processes will execute, as scheduled at this moment.
View / Now	now	Prints the current time in the system.

Table 10: Useful Simulator Button /Menu Commands

Group / Button Menu / Choice	Textual command	Action
View / Input Port	list-input-port	Shows the input port for the current process scope.
View / Timer List	list-timer	Prints the system timer queue.
View / Variable	examine-vari- able	Prints the value of a specified variable for the current process scope.Expected parameters: var_name. NOTE: the variable name can be omitted, in that case all variables of the process will be printed.
Change / Variable	assign-value	Changes the value of a specified variable for the current process scope.Expected parameters: var_name value.
Show / Coverage	print-coverage- table	Prints coverage statistics for the simulation so far. Expected parameters: file_name
Trace / Text Level: Set	set-trace	Sets the textual trace scope and level. Expected parameters: sdl_scope_entity scope_name range_value Range_value is 0..6. Example: se-tr Block Control 6
Trace / SDL Level: Set	set-gr-trace	Sets the SDL trace scope and level and starts the SDL trace. Expected parameters: sdl_scope_entity scope_name range_value Range_value is 0..2. Example: se-gr-tr System Acc 0
Trace / MSC Level: Set	set-msc-trace	Sets the MSC trace scope and instance level. Expected parameters: sdl_scope_entity scope_name range_value Range_value is 0..3. Example: se-msc-tr Block Control 2
Trace / MSC Trace: Start	start-interac- tive-msc-log	Sets the MSC trace level and starts interactive MSC trace. Expected parameters: range_value Range_value is 0..2. Example: sta-int-msc-log 2
Log / Start Complete Log	log-on	Enables logging of all the interaction between the Simulator and the user that is visible on the screen. Parameters: <Optional file name>

Table 10: Useful Simulator Button /Menu Commands

Group / Button Menu / Choice	Textual command	Action
Log / Stop Complete Log	log-off	Turns-off the interaction log facility.
File Menu / Exit	quit	The simulation is terminated.
File / Open		Start a new Simulator.
File / Restart		Restart the current Simulator
View / Watch Window		Show the Watch window (which automatically monitors variable values during a simulation).
View / Command Window		Show the Command window (which automatically monitors the output of certain commands during a simulation).
Log / Start Com- mand Log	Command-Log- On.	Start logging the commands.

- Note that the textual commands can be abbreviated as long as they are unique:

n-t	next-transition
out-to	output-to
- SDL identifiers can also be abbreviated.
Example: **output-to KeyStroke '0' Panel** can for example be abbreviated
out-to keystr '0' pan
- The textual command line has a history which can be browsed by the up-arrow key. This can be useful for repeatedly sending similar signals from the environment.
- Command scripts to facilitate future simulations, can be created by logging of all issued simulator commands. The scripts may for example contain commands or sending signals with parameter values. A command script may later be read in and executed to repeat the same command session, see *Table 10*.
- In the Simulator UI, it is possible to customize the contents and appearance of the button modules. For example you may wish to add your own command buttons for frequently used commands, or to change or delete existing command buttons. These operations are invoked from the **Group** menu in the button modules.

The Coverage Viewer

When you have run a simulation you may want to know which part of the specification you have tested and which part you have not. This can be done with the help of the Coverage Viewer.

Choose **Coverage** under the **Show** menu to show the coverage information. The Coverage Viewer window is shown in *Figure 19*.

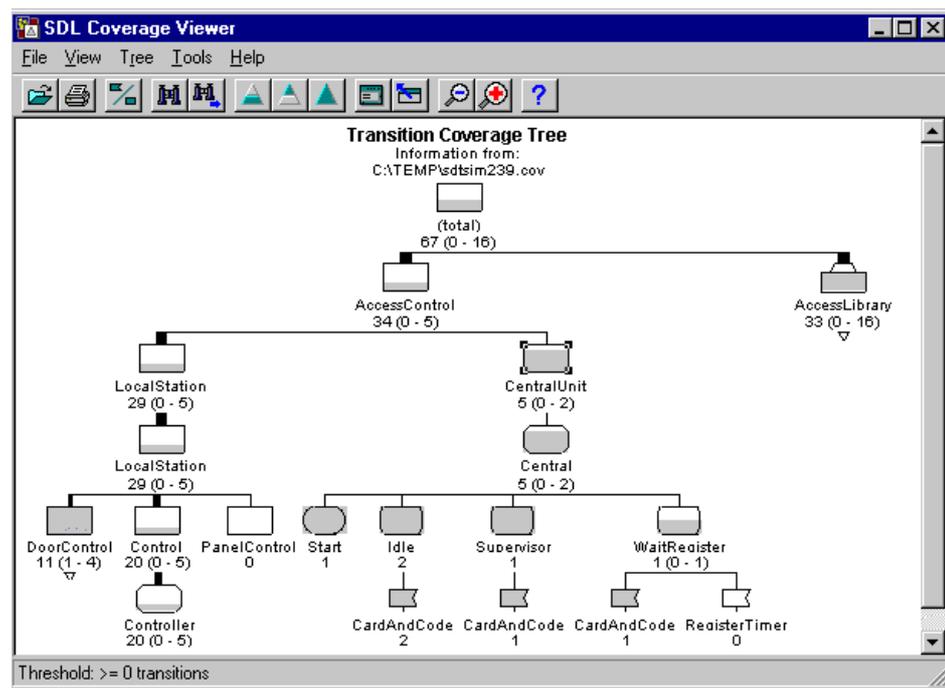


Figure 19 The Coverage Viewer window

The coverage tree can display symbols or transitions. Press the **Tree Mode** quick button to display the Transition Coverage Tree. 

In the coverage tree which will appear, you can see the number of times you have visited the different transitions (symbolized by the input symbol). If you **double-click one of the input symbols** in the coverage tree you will automatically get up the corresponding symbol selected in an SDL Editor.

You can also set thresholds on what you want to see in the coverage tree. To set that you only want to see the parts which never have been executed choose the **Set Visibility...** command under the **Tree** menu in the Coverage Viewer. Make sure that the radio button **Show transitions executed <= threshold** is selected. If not, select it and press the **Set** button.

Another way to easily change the representation is by the quick buttons: **Least**, **Most** and **All Nodes**. 

The Validator

The Validator provides an automated fault detection mechanism that checks the robustness of the application and finds inconsistencies and problems in an early stage of development. This is often referred to as verification of an SDL system.

When comparing the system against requirements, the Validator provides a possibility to perform automatic validation of the requirements expressed using the MSC notation.

If you want to create a Validator, you can use one of the following ways:

1. Select the system in the Organizer. Choose the **Make...** command in the **Generate** menu. Set the **Standard kernel** option to one of the **Validation** options in the dialog which appears. Press the **Set** button. Press the **Validate** quick button in the Organizer.
- Choose the system you want to create a simulator for in the Organizer and then choose the **Make...** command under the **Generate** menu in the Organizer. Set the **Standard kernel** option to **Validation** in the dialog which appears and then press the **Make** button. When you have created a validator you can start the validation by choosing the **Validator UI** command in the **SDL** sub-menu in the **Tools** menu in the Organizer.

The Validator UI window appears, see *Figure 20*.

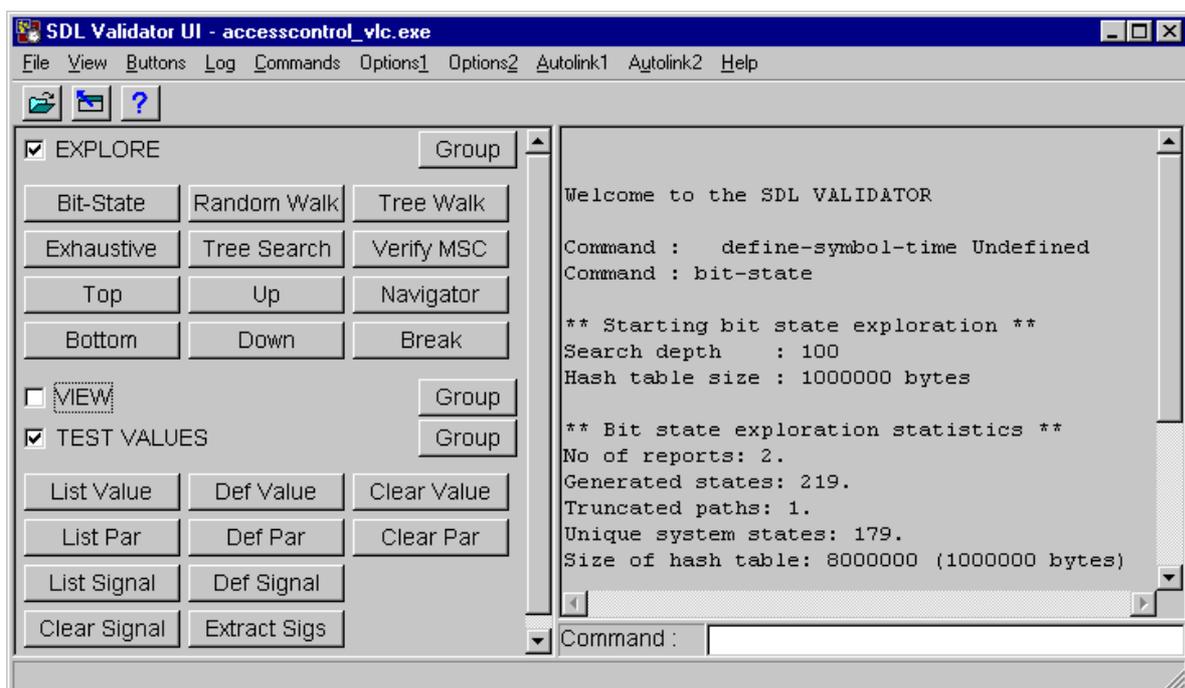


Figure 20 The Validator window

Table 11: Useful Validator Button Commands

Group / Button	Textual command	Action
Explore / Bit-State	bit-state	Starts a bit-state exploration.
Explore / Random Walk	random-walk	Starts a random walk exploration.
-	exhaustive-exploration	Starts an exhaustive exploration
Explore / Verify MSC	verify-msc	Starts an MSC verification exploration.
Explore / Break	<return>	Interrupts the current exploration.
Test Values / Def signal	define-signal	Defines an incoming signal that should be tested.
Test Values / Clear signal	clear-signal	Clears all possible signals for a signal with a given name.

Table 12: Useful Validator Menu Commands

Menu / Command	Action
File / Open	Start a new Validator.
File / Restart	Restart the current Validator
View / Watch Window	Show the Watch window (which automatically monitors variable values during a simulation).
View / CommandWindow	Show the Command window (which automatically monitors the output of certain commands during a simulation).
Log / Start Command Log	Start logging the commands.
Commands / Toggle MSC Trace	Sets the MSC trace on/off.
Commands / Toggle SDL Trace	Sets the SDL trace on/off.
Commands / Show Coverage Viewer	Opens the Coverage Viewer with the coverage results from the last exploration.

Report viewer

Information about the exploration results will be presented in the **Report Viewer** window, see *Figure 21*

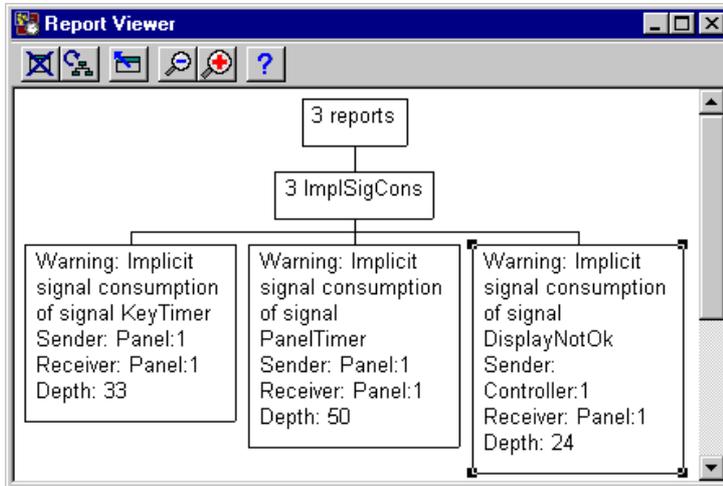


Figure 21 Validator Report Viewer window

Double-click on the box in the Report Viewer. You will now get a more detailed description of the problems. **Double-click** on the left one of the **report description boxes** you just got up. The Validator will now go to the place where the error occurred and the MSC Editor will pop up and show what happened in the system up to the point where the error occurred.

Table 13: Some Validator reports

Report name	Description
ImplSigCons	Implicit signal consumption. A signal was sent to a process that was not able to handle (or save) the signal in the current state, so the signal was implicitly consumed.
Output	A signal was sent using a PId value that did not refer to an existing process or a signal was sent (without PId) and no possible receiver was found.
Deadlock	All processes are waiting for some other process to act, implying that none of the processes will execute, no matter if signals are sent in to the system from the environment.
MaxQueue-Length	Max input port queue length exceeded. The length of the input port has exceeded the value defined by the command Define-Max-Input-Port-Length

You can also go from the MSC trace to the SDL definition. Select a symbol in the MSC diagram and choose **Show SDL Symbol** on the right mouse button.



P.O. Box 4128, SE-203 12 Malmö, Sweden · Phone: +46 40 17 47 00 · Fax: +46 40 17 47 47
Offices in Europe, USA, Asia and Australia · Distributors worldwide
info@telelogic.com · www.telelogic.com