Simulation System SIMPLORER® 6.0

User Manual

English Edition

Notice

Ansoft Corporation
Four Station Square
Suite 200
Pittsburgh, PA 15219
(412) 261 - 3200

This Manual was created with Adobe FrameMaker.

The pictures were created with Paint Shop Pro.

Printing History

New editions of this manual include material updated since the previous edition. The manual printing date, indicating the manual's current edition, changes when a new edition is printed. Minor corrections and updates incorporated at reprint do not cause the date to change.

Edition	Date	Software Revision
1	June 1998	4.0
2	January 1999	4.1
3	January 2000	4.2
4	November 2001	5.0
5	October 2002	6.0

Welcome to the SIMPLORER Simulation Center, the integrated simulator for complex technical systems. Whether you are a beginner or an experienced professional, SIMPLORER's comprehensive suite of tools gives you expert and reliable results in very little time.

You can create related simulation models quickly, process simulations accurately and reliably with the simulator backplane technology, and present and arrange the results with powerful post processors. You can also transfer the simulation data and result presentations to other applications.

This manual is designed to be used both as a command reference and as a user's guide for day-to-day tasks. SIMPLORER is continuously being developed and new functions are added regularly; hence, there may be slight inconsistencies in the documentation.

Table of Contents

General Description of Content

1 Introduction

This chapter represents general hints for using this manual, for installing SIMPLORER, and for working with graphical user interface.

2 Project Management

This chapter represents SIMPLORER's program structure and the description of global program setting (start programs, manage projects, settings for the program environment).

3 General Modeling Functions

This chapter represents the modeling with graphical input tool Schematic, including common SML conventions (names, numeric data, qualifier, expressions, predefined variables, standard functions), creating simulation models, using characteristics and subsheets, modifying model parameters, and coupling to Mathcad.

4 Modeling with Circuit Components

This chapter represents all components of the Circuit (passive elements, sources, switches, semiconductors device level, electrical machines, transformers) and all of their parameters. At the beginning you can find the reference arrow system and invalid configurations.

5 Modeling with Block Diagrams

This chapter represents all block components (continuous blocks, discrete blocks, source blocks, signal processing and math blocks) including all their parameters are described. At the beginning you can find common rules for connecting of transfer blocks, defining signal direction and defining block sample times.

6 Modeling with State Graphs

This chapter represents all state graph components (transition and states) including all their parameters and action types. At the beginning you can find the use of animated symbols.

7 Using Measuring Instruments

This chapter represents all measuring instruments (electrical, fluidic, magnetic, mechanical, thermal domain) and their reference arrow system. At the beginning you can find some typical cases of use.

8 Using Signal Characteristics

This chapter represents all components of the Signal Processing folder (general parameters, dynamic behavior parameters, dynamic performance parameters, special waveform parameters) and all of their parameters. At the beginning you can find some typical cases of use and features of the components.

9 Using Modeling Tools

This chapter represents all components of the Tools folder (time functions, characteristics, equations) and all of their parameters. At the beginning you can find some typical cases of use and features of the components.

10 Physical Domains

This chapter represents all components of the Physical Domains folder (electrical, fluidic, magnetic, mechanical, thermal domain) and all of their parameters. At the beginning you can find the reference arrow system of all domains.

11 Schematic Environment

This chapter represents the working of the model sheet itself, such as representation of results, data export, text integration, using of templates,.... In addition, the updating of model sheets from former versions are represented.

12 Simulation Results

This chapter represents the representation forms of Simulation results. This covers a common overview about data formats and the using of Display Elements and the View Tool. Both applications and their capabilities to show and represent simulation data are described in detail.

13 Simulator

This chapter represents the description of simulator functions including simulation parameters, managing the simulator queue and the description of the simulator backplane.

14 Data Evaluation

This chapter represents both SIMPLORER Post Processors and their utilities are described: DAY, for evaluations of simulation data with mathematical and graphical tools and the creation of presentations; and DAY Optim, to interpret experiments.

15 Model Libraries

This chapter represents the Model Agent, an application to maintain all libraries and models included. The following features are represented: Viewing model libraries; Copy, move and delete models; Search models; Create model libraries; and Insert new components.

16 Symbol Editor

This chapter represents the Symbol Editor, an application to modify existing model symbols, such as the default symbol of a user-defined macro. In addition, the features Animated and Interactive Symbols are described.

17 Text Editor

This chapter represents the method of creating simulation models with the Text Editor. The model is entered and simulated in the Text Editor. You cannot transfer a text model into a graphical, which is used in the Schematic. Also included in the chapter is the SPICE converter, a semiautomatic conversion tool.

18 Appendix

The Appendix offers a SIMPLORER Glossary with the most frequent terms. Furthermore a table of SIMPLORER Libraries, and tables with common Conventions are included in the appendix. Also you will find help for common problem Troubleshooting.

19 Literature References

The chapter represents a list of used literature references and their location in the User Manual.

20 Index

The chapter represents a comprehensive listing of keywords and associated information used in the User Manual.

1 Introduction

SIMPLORER is a software package use to design and analyse complex technical systems. Simulation models can contain electric circuit components of different physical domains, block elements, and state machine structures. SIMPLORER's simple graphialc interface makes even complex models easy to define. Fast and stable simulation algorithms reduce design time and provide reliable results.

The various tools used for modeling, simulating, and analyzing are integrated within the SIMPLORER Simulation Center (SSC). The SSC Commander starts the programs, manages the project files, and sets options for both simulation and program environment.

SIMPLORER Documentation

The following guides and manuals quickly direct you to the help you need while working with SIMPLORER.

Manuals

Getting Started — Program functionality at a glance, new functions, step by step simulation examples, program conventions.

User Manual — Program functions, SIMPLORER's environment settings, and detailed descriptions of internal SIMPLORER components.

Additional Modules — Description of additional SIMPLORER modules such as Couplings, C-interface, and optimization algorithms.

Installation Guide — Requirements for network version, hardware requirements, driver for hardware key, and network sharing.

Online Help

Online Help — Online help is available for all components and SIMPLORER programs, with index and keyword catalogs.

There is also a description of additional SIMPLORER modules such as Couplings, C-interface, and optimization algorithms.

Examples — For each basic component is an example available. The example can be opened with the function **«Example»** in the component's shortcut menu.

Online Documentation

Manuals — Manuals in PDF Format for printing.

About this User Manual

User Manual Conventions

SIMPLORER is fully Windows compliant. The operation of the windows and the environment conditions are consistent with the Windows standard, and therefore, are not described in this User Manual. If you have questions about Microsoft Windows, please consult your Windows documentation.

The following format conventions are used:

<Apply>	Button to confirm selection activity.
FILE>OPEN	Menu sequence to start an action.
«Properties»	Text on menus and option fields.

In many cases, clicking the right mouse button displays a **shortcut** menu with special object activities or properties. If you want to see the choices on the shortcut menu, try the right mouse button. When you see the term *mouse click*, always use the **left** mouse button.

User Manual Symbols

☑ ☐ ⊙ ○	Indicates the state of check boxes and option buttons in the text.
	Indicates a helful hint for using a feature.
	Indicates important information – Please note!
	**Content signpost** Indicates detailed specifications of content at the beginning of chapters and sections.
	**Function sequences** Indicates the beginning of instructions that describe the procedure for a complex task.
	**Reference guide** Indicates the beginning of lists and tables that contain complete summaries of menu commands and parameters.
	**Basic knowledge** Indicates general descriptions and keys points to understanding program functions.

Installing SIMPLORER

Hardware and Software Requirements

CPU: Pentium 600 MHz
Memory: 256 MB
Free Hard disk: 2 GB
Operating system: Windows 2000/Windows NT 4.0/XP
Graphic card: VGA 1024x768

The complete installation needs approximately 150 MB memory of hard disk space; however, the output of simulation data may require much more hard disk space.

See also the separate Installation Guide.

Contents of the SIMPLORER CD

Directory	Description
DaoSetup	DAO (Data Access Object) setup files
Data	Autorun files
Documentation	SIMPLORER documentations with installation guide, Getting Started Guide, User Manual, and VHDL-AMS Manual, in PDF format
FLEXlm	Setup and documentation for FLEXlm[a]
SIMPLORER6	SIMPLORER program files, model libraries, examples, and DAO setup
Tools	Additional programs – Adobe Acrobat Reader[b] as setup and executable program from CD. – HTML workshop compiler for creating own help files.

a. FLEXlm is a software for electronic licensing to control the use of software from GLOBEtrotterSoftware Inc., San Jose, CA 92125 USA.
b. Adobe Acrobat Reader is software for viewing documents in PDF-format (Portable Document Format) from Adobe Systems Inc., 345 Park Avenue, San Jose, CA 95110-2704 USA.

Installation under Windows 2000 and Windows NT

Before starting every installation, make sure that the hardware key is connected to the parallel printer port (LPTx) of your PC.

To install under Windows NT, you must log in as an administrator.

Basic Installation Procedure

1. Copy the **license.dat** file (usually on a separate floppy disk or in e-mail) for the purchased software on your PC.
2. Install the hardware key on the parallel port of the computer, if it is included with your installation. The hardware key must always be installed when SIMPLORER is used.
3. Log in as Administrator for Windows NT/2000 systems. (Otherwise, you will see error -115 when files are being copied.)
4. Run the FLEXlm setup (once for the whole network, usually on the server or, on the single machine if you install a single machine version).
5. Run the SIMPLORER setup.

Starting Setup

1. Put the SIMPLORER CD-ROM into the CD-ROM drive. The setup program starts automatically.
2. To start the installation, click <Installation>.
3. Follow the instructions in the setup program.

If the setup program does not start automatically, use Windows Explorer to open the file *Autorun.exe* on the SIMPLORER installation disk.

2 Project Management

With SIMPLORER's file management system, all files from a simulation task are collected in a project file (.ssc file). When you open a project, all the files from the simulation task are available, and you can edit and start any of these files in the SSC Commander (SSC: SIMPLORER Simulation Center). The SSC Commander also controls user settings, SIMPLORER program settings, and the simulator queue.

SIMPLORER project files manage files of:

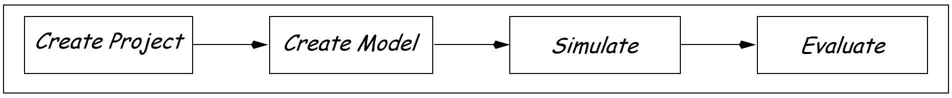
- Models (.ssh, .sml)
- Characteristics (.mdx, .mda, .mdk)
- Results (.sdb)
- Presentations (.day)
- Experiments (.exp)
- Model libraries (.smd)
- External programs (.doc, .txt, …)

This chapter contains information on:

- Steps to solve a modeling task
- SIMPLORER's program structure
- Programs and model libraries of SIMPORER's basic version
- Program management
- Starting SIMPLORER
- Starting SIMPLROER programs
- Managing project files
- Starting simulations
- SIMPLORER files and their programs
- Defining global settings and options for SIMPLORER programs
- Creating a file with version information

2.1 Basic Sequence from Simulation Problem to Result

Solving a modeling task with SIMPLORER consists essentially of four steps. The figure below shows these steps, from creating the new project to evaluating the results. If the problem is complex and comprehensive, individual steps may need to be repeated.



Create project: A project is a file that contains the different files of a simulation task. The SSC Commander both creates and manages these project files.

Create Model: A model is required to start a simulation. Use the graphical input tool Schematic or the SIMPLORER Text Editor to create a model.

Simulate: The simulator calculates the simulation model and sends the results to a display.

Evaluate: Simulation results can be evaluated and analyzed using the DAY Post Processor, SIMPLORER's program for analyzing simulation data.

2.2 SIMPLORER's Program Structure

The various program tools for modeling, simulating and analyzing are integrated in the SIMPLORER Simulation Center (SSC). The SSC Commander starts programs, manages project files and determines options for the simulation and program environment.

Basic Version and Optional SIMPLORER Programs

This User Manual describes the programs and libraries in the Basic version. SIMPLORER is continuously being developed and new functions are added regularly; hence, there may be slight inconsistencies in this user manual.

<u>Basic Version</u>

- Graphical Input Schematic with Mathcad[1] interface
- SIMPLORER Text Editor
- Model database Model Agent
- DAY Post Processor with Mathcad and Matlab[2] interface
- DAY Optim Post Processor
- Experiment tool with Multi simulation and Trend analysis
- Basics model libraries and standard libraries

1. Mathcad is software from MathSoft, Inc. 101 Main Street, Cambridge, MA to compute mathematical problems.
2. Matlab is software from Mathworks, Inc., 24 Prime Park Way, Natick, MA.

Opional Programs

- Analytical Frequency Analysis (Computation of transfer functions)
- Simulative Frequency Analysis
- Experiment tool with Optimization

Interfaces

- ECE (equivalent ciruit extraction) Maxwell[1] interface (interface.smd)
- RMxprt[2] interface (interface.smd)
- Full-Wave SPICE interface (interface.smd)
- Mathcad interface (interface.smd)
- Matlab/Simulink[3] interface (interface.smd)*
- C interface for linking user defined models in C code*
- IEEE interface*

Add On Model Libraries

- Electric power components (power6.smd)*
- Set of components suited for needs of the automotive industry (automotive6.smd)*
- One-Dimensional Mechanical System Module (mechsim6.smd)*
- Hydraulic components (hydraulic6.smd)*
- Library with machine models (*_ist6.smd)*
- Nonlinear transmission and linear transmission components (transfer6.smd)

*These libraries are optional (not included in the basic SIMPLORER package). Optional programs and libraries can be delivered on request. Please contact your Ansoft sales representant for the current prices.

SIMPLORER Version Sizes

The table shows the maximum number of components available in the SIMPLORER versions.

Module	Description	Student	University
Conservative nodes	Circuit notes	no limit	no limit
Circuit	Basic circuit components	20	45
Block diagram	Basic block components	15	30
State graph	Transfer conditions	30	40
C interface	DLLs to integrate models	0	0
Subsheets	Embedded substructures in Schematic	1	1
Text macros	Components modeled as macro	no limit	no limit
Display Elements	Elements for graphical output on sheet	4	4
Drawing elements	Drawing elements in Schematic	no limit	no limt

1. Maxwell 2D/3D is a software from Ansoft Corp., 4 Station Square, Pittsburgh, PA to solve two-dimensional and three-dimensional electromagnetic problems.
2. RMxprt is a software from Ansoft Corp., 4 Station Square, Pittsburgh, PA to design rotating electric machines.
3. Matlab/Simulink is software from Mathworks Inc., 24 Prime Park Way, Natick, MA.

2.3 Program Management with the SSC Commander

The SSC Commander is SIMPLORER's control center, where program modules are started, projects are managed, and options for the program environment are determined.

<u>This section contains information on:</u>

- Starting SIMPLORER and SIMPLORER programs
- Managing projects and project files
- Defining settings for SIMPLORER programs

Starting SIMPLORER and Defining User Names

1 Start SIMPLORER from the Windows taskbar:
Start/Programs/Ansoft/SIMPLORER 6.0/SIMPLRORER Simulation Center 6.0

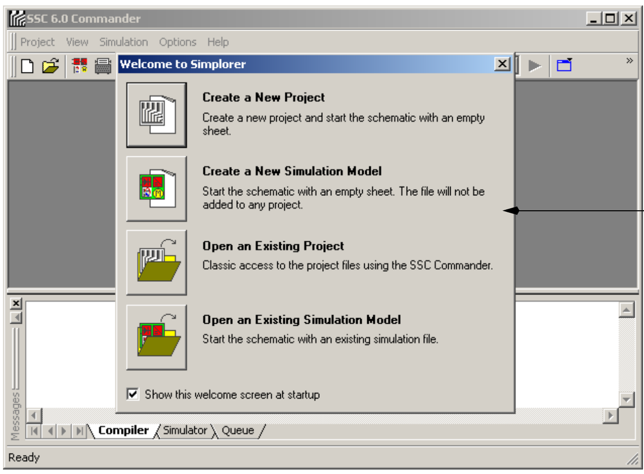
2 Select a user name from the **«User»** box. Open the drop-down box to see the list of user names. By default, immediately after the SIMPLORER installation, the box displays the current Windows user name.

New user names: Enter the new user name into the box. A special profile (for example directories, screen layout) is connected with each user name in the SIMPLORER program environment.

There are two ways to start SIMPLORER from the start-up-screen: Click <OK> or click <Last Project>. <OK> opens the SSC Commander without a project; <Last Project> opens the last project with all its files.

Starting SIMPLORER without a Project

From the SIMPLORER start-up-screen start the SSC Commander by clicking <OK>. The Welcome screen offers four choices to start your work: create or open a project; and create or open a simulation model.



SIMPLORER Welcome Dialog

The **«Show this Welcome Screen at Startup»** option determines whether the welcome dialog is displayed each time the program starts. In OPTIONS>COMMANDER SETTINGS you can set the option to display or hide the Welcome Screen.

Creates a new project (all files created during a simulation task are included in that project) and starts the Schematic with an empty sheet. Creating a new project is best if a simulation model contains characteristics, evaluation files, special libraries or other documents. A project can also be created with the PROJECT>NEW command. However, only one project can be open at a time.

Creates a new simulation model by starting the Schematic with an empty sheet. When you create a new model, the files you create will not be included in a project automatically. Working without a project is useful if you want to open only a Schematic file. You can include the file in a project later.

Starts the Open dialog for an existing project. All files included in the project can be started directly from the SSC Commander.

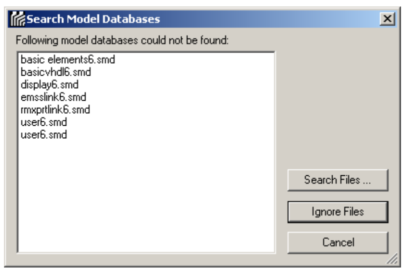
Starts the Open dialog for an existing simulation model. No project is opened or created.

Starting SIMPLORER with a Project

From the SIMPLORER start-up-screen click <Last Project>. The SSC Commander opens the last project. All the files from the last simulation task are available, and you can start and edit the files collected in the project file in the SSC Commander.

Errors when the Program Starts

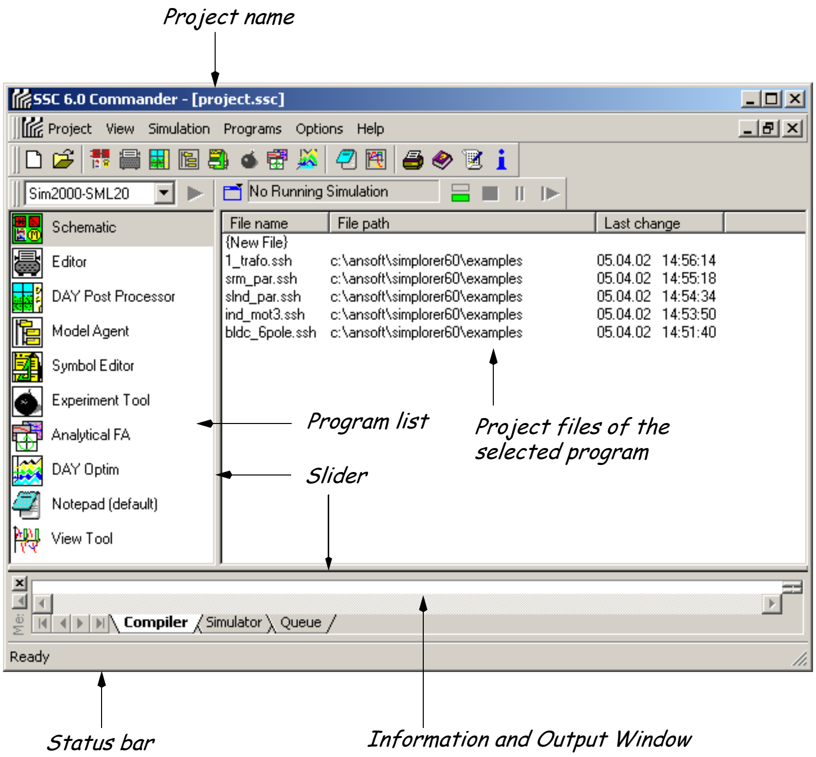
If model libraries are not found when the program starts, you will see an error message like the one in the figure below. If you click <Search Files…>, a dialog to search new model libraries openes. <Ignore> skips all missing files, <Cancel> skips all missing files too, but a error message appears if you want open a missing file.



Error message if files were missing during the program start

2.4 SSC Commander Main Window

After loading an existing project or creating a new project, you will see a window similar to the one below. To create a new project file of the program type selected in the list on the left-hand side, double-click the entry {New File} in the project file list.



During a simulation task, the Information window displays program messages, error messages, and reports from the compiler, simulator, or simulator queue. You can change the window's layout by moving the sliders with the mouse.

View Menu

Command	Toolbar Symbol and Description	
MESSAGES	✓	Shows or hides the Information window.
STANDARD TOOLBAR	✓	Shows or hides the default toolbar.
SIMULATION TOOLBAR	✓	Shows or hides the simulation toolbar.
STATUS BAR	✓	Shows or hides the Status bar.

Starting SIMPLORER Programs

SIMPLORER programs can be started with the symbols located in the standard toolbar, the PROGRAMS menu or with the symbols in the SIMPLORER program list.

If both a program (Schematic) and a file are selected, a double-click starts the respective program and loads this file. If both an program and {New File} are selected, a double-click starts the program with an empty window.

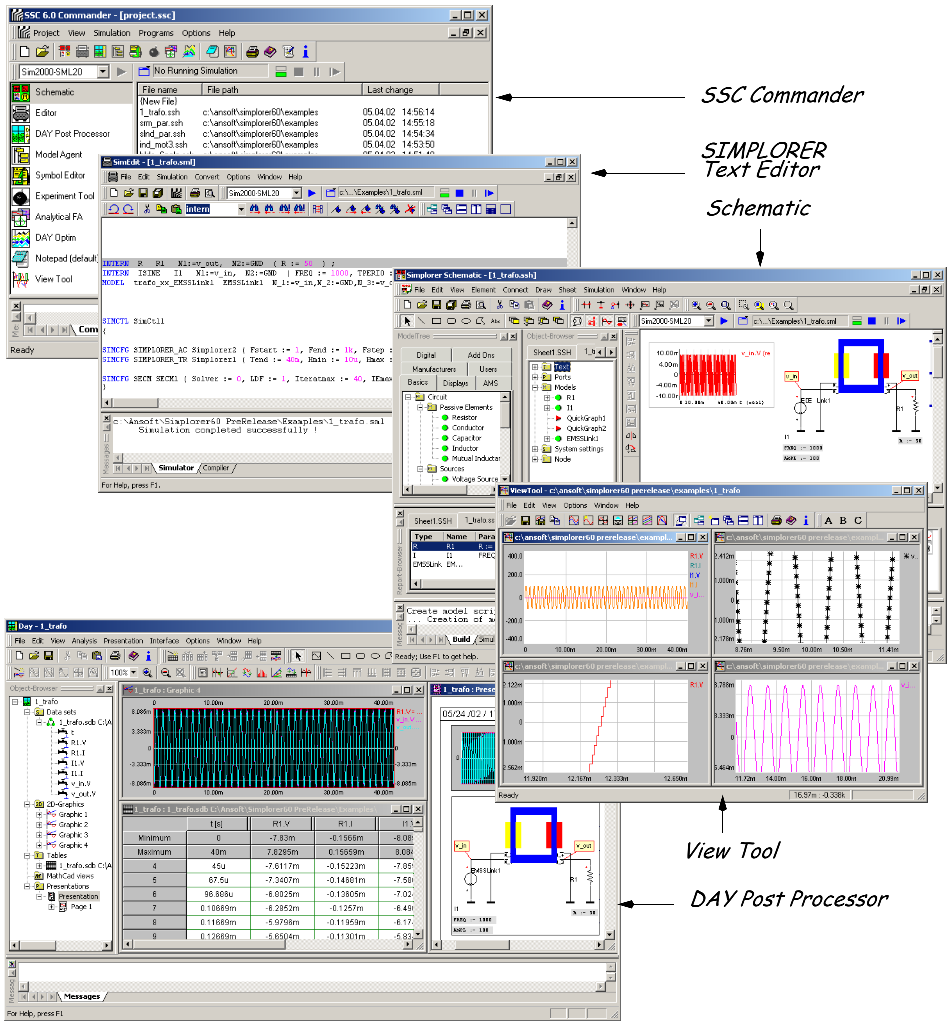
The program list is user-defined and can be changed in the SSC Commander OPTIONS>COMMANDER SETTINGS menu.

The SIMPLORER Program List

Programs Menu ▾

Command	Toolbar Symbol and Description
SCHEMATIC	Schematic graphical input tool used to create simulation models.
EDITOR	SIMPLORER Text Editor used to create simulation models in the SIMPLORER Modeling Language (SML).
MODEL AGENT	Model Agent used to manage model libraries and macros.
SYMBOL EDITOR	Symbol editor used to edit the symbols used in the Schematic to represent models (components).
EXPERIMENT TOOL	Experiment tool used to create and manage simulation sequences.
DAY POST PROCESSOR	DAY Post Processor used to analyze and process graphic simulation results.
DAY OPTIM	DAY Optim Post Processor used to analyze optimizing sequences.
VIEW TOOL	View Tool to display an existing simulation result.
ANALYTICAL FG	Analytical Frequency Analysis to determine the frequency response of known transfer functions.
NOTEPAD (DEFAULT)	External editor used to process ASCII files.

Open SIMPLORER Programs

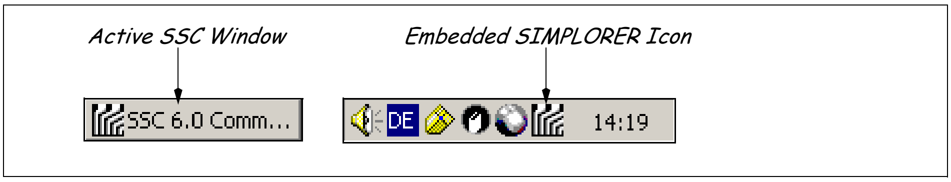


The SSC Commander starts all SIMPLORER programs and manages the project files included with each program. If you close the SSC Commander, all other SIMPLORER programs will also close.

SSC Commander Icon on the Taskbar

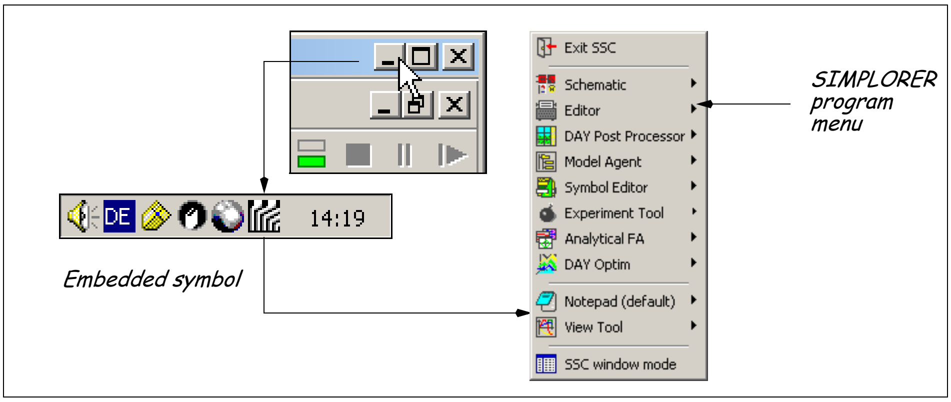
After the program starts, the SIMPLORER icon appears on two places in the Windows taskbar:

- as a program symbol next to the other active programs
- as an embedded icon next to the time and date display



SSC Commander in the Background

If you start SIMPLORER programs or other applications without minimizing the SSC Commander, the open SSC window is placed in the background. Along with the embedded symbol, the symbol for the active window is visible on the Windows taskbar. Click one of these symbols on the taskbar to open the SSC Commander on the SSC window mode.



If the SSC Window is minimized, click the embedded icon to open the PROGRAM menu to start SIMPLORER programs. (Set this option by choosing this command sequence from the SSC Commander: OPTIONS>COMMANDER SETTINGS: ☑ **«Hide program icon when minimized»**.)

If the Windows taskbar is invisible, you can see it by moving the mouse pointer over the screen border.

If the **«Hide program icon when minimized»** option is deactivead, the program icon is also visible on the taskbar after minimizing the SSC Window.

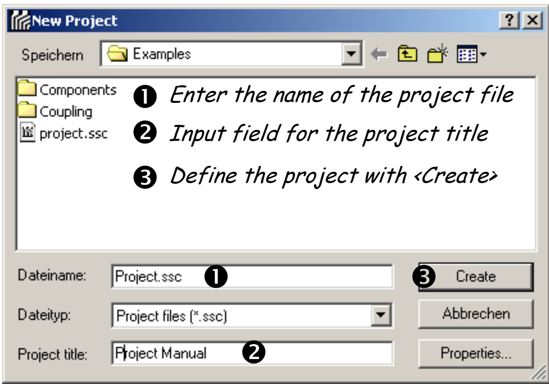
2.5 Project Management with the SSC Commander

All files representing a simulation task (SML source code, characteristic functions, simulation data, etc.) belong to a project. Files can be assigned to a project when saving the files for the first time or later the PROJECT menu.

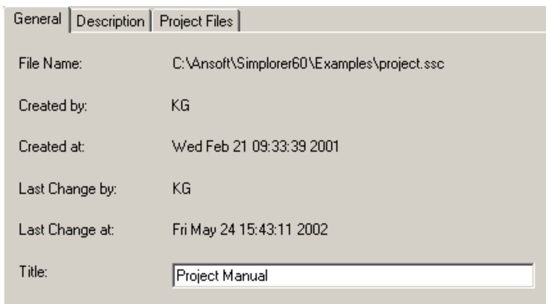
Only one project can be using open at a time.

Project Menu

Command	Toolbar Symbol and Description
NEW	Creates a new project.



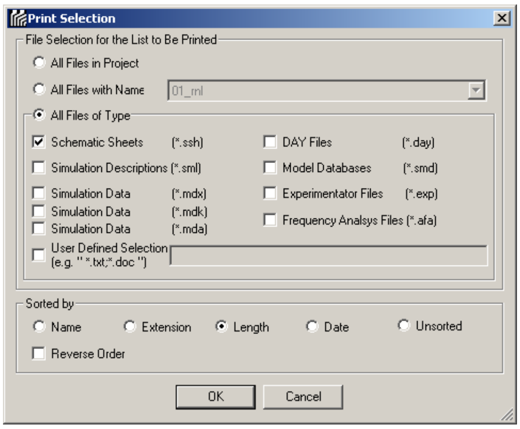
Command	Toolbar Symbol and Description
OPEN	Opens an existing project.
COPY	Copies all files belonging to a project to a new location on any storage system.
EDIT	Modifies the title, the description and the files belonging to an existing project.



Menu to edit project properties

Command	Toolbar Symbol and Description
CLOSE	Closes a project.

Command	Toolbar Symbol and Description
PRINT	Prints a list of files belonging to a project. The first opened print selection dialog sets the file and sorting criteria for the list. The page layout is defined in OPTIONS>COMMANDER SETTINGS «Print page».

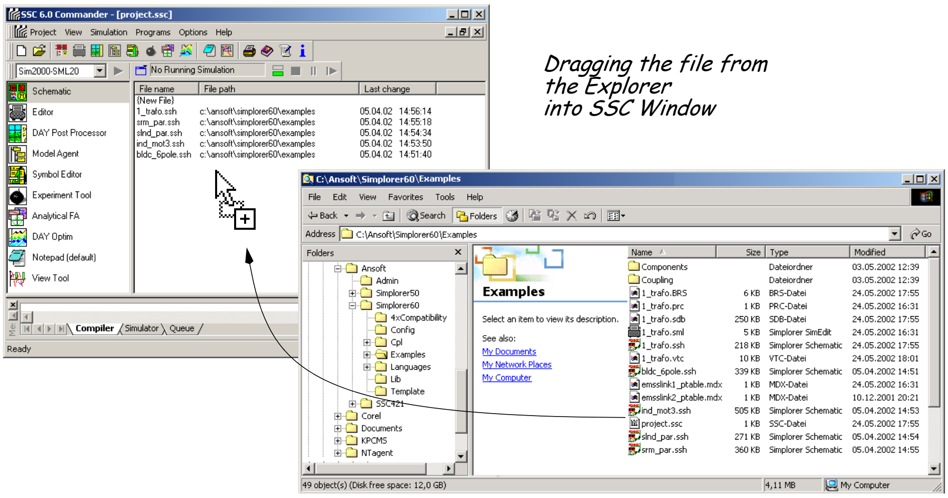


Print selection dialog

Command	Description
PRINT PREVIEW	Displays the print page with the list of the project files. The first opened print selection dialog sets the file and sorting criteria for the list.
PRINTER SETUP	Sets the printer options: paper size, paper feed, and format (portrait or landscape).
SEND AS E-MAIL	Creates a new e-mail message, and attaches the selected project files. You can also set an option to zip the files. You need an email program on your PC to use this function.
«FILE LIST»	Lists the last opened files.
EXIT	Closes the SSC Commander and included programs.

Drag-and-Drop from Windows Explorer

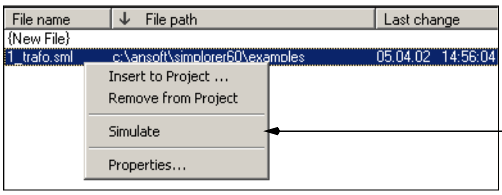
Any file located in the Windows Explorer can be integrated into the active project using drag-and-drop. To move a file into the active project, click the file, hold the mouse button, and move the mouse pointer into the project file list. In a valid insert area, the mouse pointer appears as a rectangle. The SSC Commander displays available files depending on the selected program and the file extension setting in OPTIONS>PROGRAM DIRECTORIES «SIMPLORER Modules».



Dragging the file from the Explorer into SSC Window

Editing a File List from the Shortcut Menu

Besides using the SSC Commander menu, you can also manage project files from the shortcut menu, accessed by clicking the right mouse button.



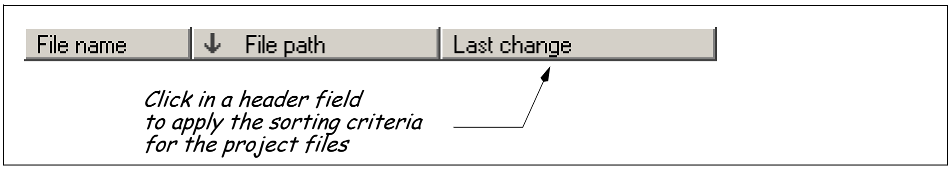
Shortcut menu for a project file

Commands on a file’s shortcut menu

Insert to Project	Inserts additional files.
Remove from Project	Removes a selected file from the project.
Simulate	Simulates files with the extension .sml.
Properties	Displays and changes file properties.

Sorting Criteria

The sorting criteria are located in the header of the file list. By default, the entries are sorted in alphabetical order. Click a header field to apply a new sorting criterion; click again to perform the sort.



SIMPLORER File Structure

SIMPLORER creates and uses several program-specific files. The following table gives an overview of the created and used files and their corresponding programs.

Program	File	Description	Created (c) or used (u) data files
SSC Commander	Project.ssc	File management for a project	
Schematic	Model.ssh	Model description on different levels	Data.mdx, Data.mda/mdk, Data.xls, Data.mdb, Data.csv, Data.txt (c/u) Model.sml (c)
SIMPLORER Text Editor	Model.sml		Model.sml (c/u)
Experiment tool	Experiment.exp	Simulation sequence of experiments	Task.mdx (c) Experiment.log/rpt (c)
Model Agent	Database.smd	Model databases	Model.sml (u)
Analytical Frequency response	Model.afa	Data for transfer functions	AF_Model.mdk (c)
DAY Post Processor	Analysis.day	Analysis file with different simulation results	Data.sdb, txt, out (u) Data.mdx, mdk/mda, xls/csv, mdb (c/u)
DAY Optim	Data.mdk/mda/mdx	Data files	Data.mdx, mdk/mda (c/u) Filter.fil (c/u)
Simulator		Model description Initial values State values Data files Experiment files	Model.sml (u) Model.aws (c/u) Model.krn (c/u) Data.mdx, mdk/mda (c/u) Model.brs (c)

The following files are created after a simulation starts in the SSC, Editor, Schematic, or Experiment tool:

- Data: Data.sdb, Data.mdx, Data.krn, Data.aws, Data.brs, Data.prc, Data.vtc
- Temporary: … Model_SML\Model.smt

You can delete temporary files when you close the SSC Commander. See also “Settings and Options for SIMPLORER Programs” on page 18.

Starting Simulations

In the SSC Commander, simulation models can be started from the file list and managed through the simulation queue. To start the simulation, you can do one of the following:

- Click and drag a project file with the extension .sml or .idl into the simulator queue of the Information window.
- Select a model file from the project, and press <F12>.
- Select SIMULATION>SIMULATE MODEL, and select a model file.

After all previous simulation tasks in the queue have been processed, the new simulation task will be started by the designated simulator. See also “Simulation in the SSC Commander” on page 409.

2.6 Settings and Options for SIMPLORER Programs

Settings and properties can be specified in the OPTIONS menu.

From the OPTIONS menu you can,

- Add new files to the project.
- Delete temporary files when the program is closed.
- Configure the program list.
- Define user management settings.
- Define program paths for SIMPLORER.

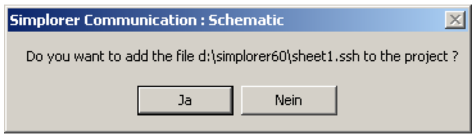
All settings are saved automatically when you close the OPTIONS dialog.

General Settings

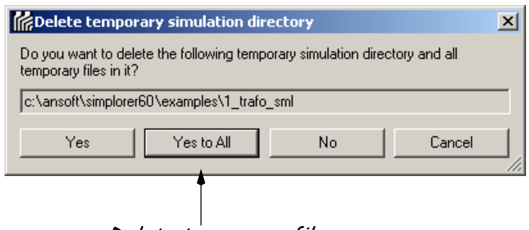
Options Menu — SSC Commander Settings ▾

Dialog Tab/Options	Description
«General» tab	
Close all documents in SIMPLORER programs at project change	☑ Open SIMPLORER files are closed. If changes have not been saved, a prompt to save them appears. ☐ Open files and also the files from the new project are available.

Dialog Tab/Options	Description
Add new files automatically to the project	☑ During the save any new files created are added automatically to the active project. ☐ The user decides if new files are added to the project or not. This dialog window appears every time new files are simulated or saved.



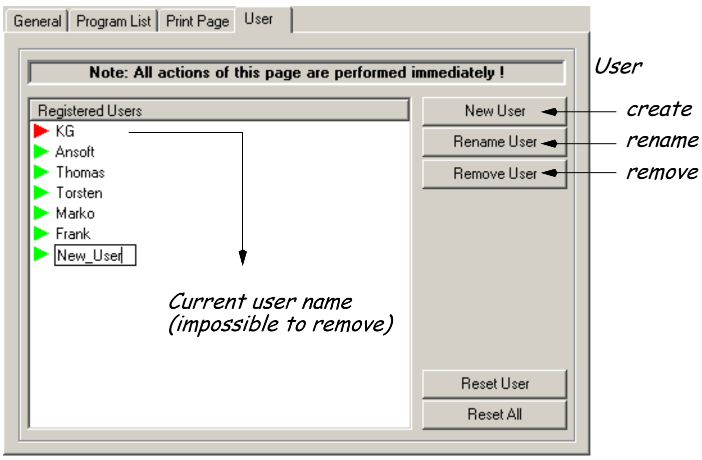
Dialog Tab/Options	Description
Hide program icon when minimized	☑ When the application window is minimized, only the embedded symbol is visible in the taskbar. ☐ When the application window is minimized, both the embedded symbol and the symbol for the active window are visible on the taskbar.
Show the Welcome Screen at Startup	☑ The Welcome Screen appears each time the program starts. ☐ The Welcome Screen is not displayed when the program starts.
Temporary files and directories at project closing	⦿ Keep: temporary files will be kept. ⦿ Confirm deletion: activates the automatic prompt to delete temporary files when a project is closed.



Delete temporary files in the file path

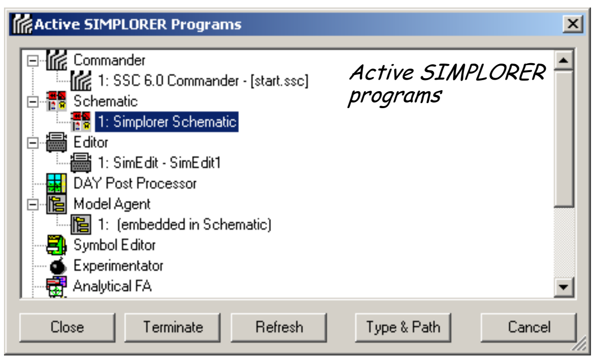
Dialog Tab/Options	Description
	⦿ Delete automatically: temporary simulation files will be **deleted** when SIMPLORER is closed. See also “SIMPLORER File Structure” on page 17.
File List	⦿ Restore column width: The user defined column width is kept. ⦿ Automatically set column width: The column width is adapted automatically.
«Program list» tab	Sets and changes the programs in the program list.
«Print Page» tab	Sets the print layout of the project list (header/footer/margin). The list shows all project files.

Dialog Tab/Options	Description
«User» tab	Shows the active user names and creates, deletes and resets user settings.



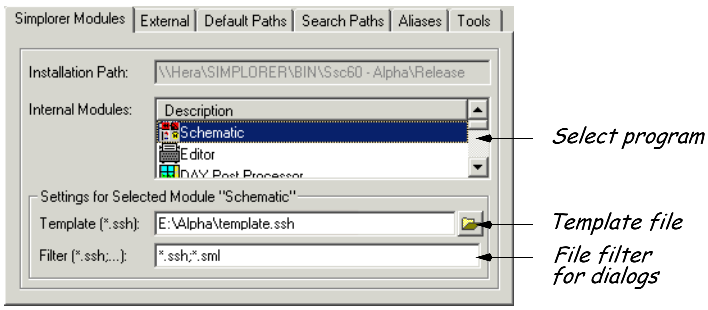
Options Menu — Active Programs

Command	Description
ACTIVE PROGRAMS	Shows the active SIMPLORER programs. Individual program modules can be closed here. This function is similar to the Task Manager in Windows.

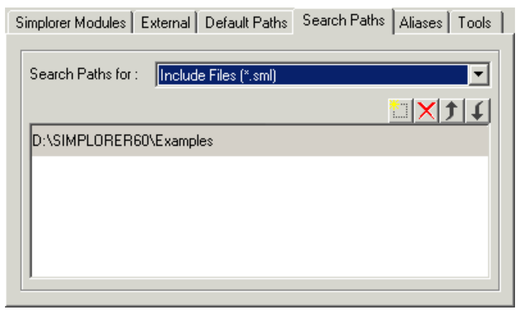


Options Menu — Directories

Dialog Tab	Description
«SIMPLORER Modules» tab	Defines Program paths, templates, and file extension filter. Templates are loaded when the corresponding program is launched.



Dialog Tab	Description
«External» tab	Paths, program names, templates, and file extension filters for external programs. External programs, included in the SSC Commander, can be started within a SIMPLORER project. If an icon is available, it is added to the program list.
«Default Paths» tab	Example path, library path, user files, and help files. File paths, defined in the Model Agent for components, are used relative to the path set in this dialog, for example, the path *C:\Ansoft\Simplorer60\Examples* in the dialog and the path and file name *Component\Circuit.ssh* as an example sheet for a component in the Model Agent results in *C:\Ansoft\Simplorer60\Examples\Component\Circuit.ssh*.
«Search path» tab	Search paths for include files (.sml), characteristics files (.mdx), and program files (.dll). These paths are tested in addition to the default paths if the corresponding file is used in a simulation description.



Dialog Tab	Description
«Aliases» tab	Defines alias names for files and paths used in VHDL-AMS descriptions.
«Tools» tab	Sets the path to the Matlab directory.

Search Paths for Files Before Simulation Run

Search Paths for Macro Models in Libraries

1. File name and path of the .smd library file in model description.
2. Paths of the current project libraries.
3. Paths of the installed libraries with the same name.
4. Paths of all installed libraries.

Search Paths for .mdx Data Files

1. File name and path in model description.
2. Path of the simulated .ssh or .sml model file.
3. Path for libraries specified in OPTIONS>PROGRAM DIRECTORIES **«Default Paths»**.
4. Subfolder */DataSet* in the path for libraries specified in OPTIONS>PROGRAM DIRECTORIES **«Default Paths»** .
5. Paths for .mdx files specified in OPTIONS>PROGRAM DIRECTORIES **«Search Paths»**.

Search Path for .dll Model Files

1. Path of the simulated .ssh or .sml model file.
2. Paths for .dll files specified in OPTIONS>PROGRAM DIRECTORIES **«Search Paths»**.
3. Path of *Windows* directory.
4. Path of *Windows\System* directory.
5. Directory defined with an environment PATH variable.

The path, defined in the **«Files»** tab of model properties, is not used in the search algorithm.

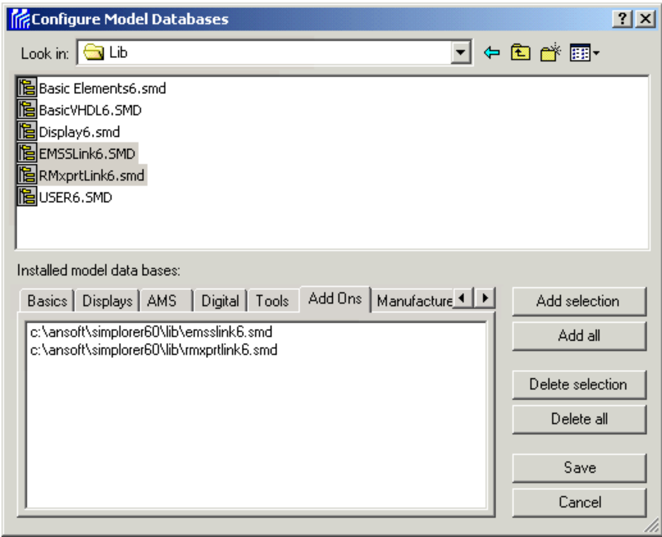
Search Paths for .sml Include Files

1. File name and path in model description.
2. Path of the simulated .ssh or .sml model file.
3. Paths for .sml files specified in OPTIONS>PROGRAM DIRECTORIES **«Search Paths»**.
4. Path for libraries specified in OPTIONS>PROGRAM DIRECTORIES **«Default Paths»**.

Options Menu — Model Databases

Command	Description
MODEL DATABASES	SIMPLORER Models are linked to model libraries (databases). After installation, the standard libraries are available immediately. User defined libraries must be installed before they can be used. In the upper window, select one or more libraries (by holding down the SHIFT key).

Command	Description
	Then click a tab («Basics», «Displays», «AMS», «Digital», «Tools», «Add Ons», «Manufacturers», or «Users») where the selected libraries should be inserted. Click the <Add Selection> button to install the selected libraries.



Be sure to validate your library installation with the <Save> button.

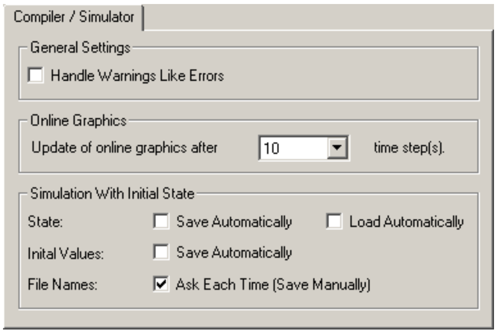
The Options menu — Language Settings

Command	Description
LANGUAGE	Defines the SIMPLORER menu language. You must restart SIMPLORER after changing the language.

Compiler and Simulator Settings

Options Menu — Simulator Settings ▾

Option	Description
«General Settings»	Settings for the compiler and simulator.



Option	Description
«Handle Warnings Like Errors»	☑ Stop the simulation at warnings. ☐ Displays warnings in the info windows and continues the simulation.
«Simulation With Initial State»	These options automatically save state and initial values of a simulation model in separate files at the end of the simulation.
«State – Save Automatically:»	☑ Saves the states of all components contained in a simulation model. You can continue a simulation later with this state file.
«State – Load Automatically:»	☑ An existing state file is loaded automatically at the simulation start and used for the simulation.
«Initial Value – Save Automatically:»	☑ Saves the values of energy storages contained in a simulation model (inductances and capacitances). These values can also be used as initial values for a simulation.
«File names – Ask Each Time (Save Manually)»	☑ Special prompt for file names, so that user-specific names can be used.

2.7 Version’s Report and Online Help

Online help and version information are both available with the SSC Commander, as for all other SIMPLORER programs, from the HELP menu.

Using Version’s Report

Version’s Report provides information about your system and the installed SIMPLORER version. You can create an ASCII file with the names of all .exe and .dll files in the SIMPLORER version. This information is important for support inquiries.

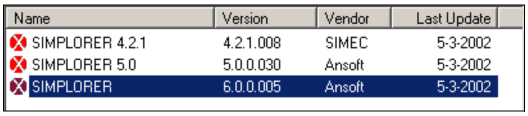
SIMPLORER Web-Update

Ansoft recommends that you download the latest SIMPLORER Product Update to ensure enhanced operation of your Ansoft products.

When you choose HELP>SEARCH FOR UPDATE, a list of new information about your SIMPLORER installations is sent from Ansoft's servers. This information is displayed in the «Applications» dialog. To check, if an update is available, select the application from the list and click <Check for Update> and follow the instructions on the screen.

To define your user-specific Web settings (connections methods and Proxy connection settings) select the **«Settings»** dialog.

To update the WebUpdate tool itself select the **«Update WebUpdate»** dialog.



Name	Version	Vendor	Last Update
SIMPLORER 4.2.1	4.2.1.008	SIMEC	5-3-2002
SIMPLORER 5.0	5.0.0.030	Ansoft	5-3-2002
SIMPLORER	6.0.0.005	Ansoft	5-3-2002

List of applications

Online Help

Online help is available for all SIMPLORER programs, from the HELP menu or by clicking the symbol on the toolbar.

Help Menu

Command	Toolbar Symbol and Description
HELP CONTENTS	Opens SIMPLORER’s online help.
ABOUT SIMPLORER	Displays information about product and version.
SEARCH FOR UPDATE	Starts the Web-Update for installed SIMPLORER applications.
VERSIONS REPORT	Creates an ASCII file with the names of all .exe and .dll files in the SIMPLORER version. This information is useful for support inquiries.

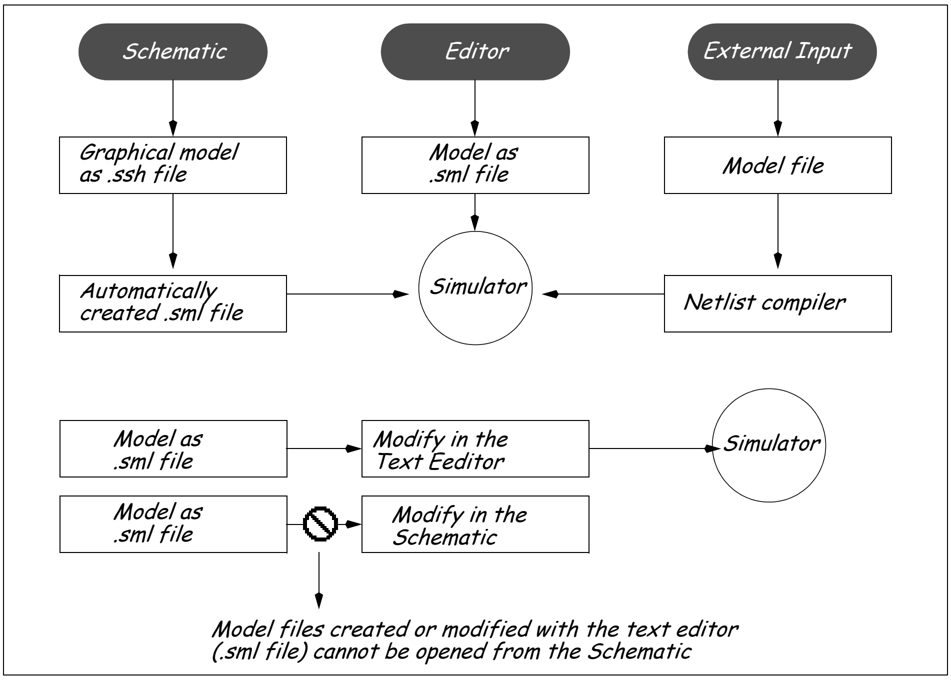
2

3 General Modeling Functions

A model description can be created in several different ways:

- Using the Schematic graphical input tool
- Using the SIMPLORER Text Editor.
- Using external graphical inputs and text editors

With the Schematic graphical input tool, you position and connect components on the sheet. With the SIMPLORER Text Editor, you create the model description in SML notation.



<u>This chapter contains information on:</u>

- Common SML conventions (names, numeric data, qualifier, expressions, predefined variables, standard functions)
- Schematic operating environment
- Creating simulation models
- Characteristics in simulation models
- Subsheets in simulation models
- Modifying model parameters
- Coupling to Mathcad

3.1 Common Conventions

The SIMPLORER Modeling Language (SML) is the general internal description language of SIMPLORER. SIMPLORER 6.0 uses Version 2.0 of SML. If a model is created in Schematic, the new SML description is automatically generated from the graphical model.

Names of Components and Variables

User-defined names can be given to all components (internal components, C models, macros, nodes, ports and variables). Names may consist of letters, digits, and underscores and can have a maximum of 50 characters.

Vowel mutations (umlauts) and spaces are not allowed. All names are case sensitive. The first character must always be a letter.

The following are **not** allowed for names:

- SML notation keywords
- Simulation parameters
- System variables

See also "Predefined Variables" on page 31.

Unit Suffixes of Numeric Data

Numeric data can be entered in Schematic component dialogs and in the Text Editor, using the following unit extensions:

Suffix	Value		SML		Examples
tera	10^{12}	E12	t	TER	*5e12, 5t, 5ter*
giga	10^{9}	E9	g	GIG	*1.4e9, 1.4g, 1.4gig*
mega	10^{6}	E6		MEG	*-1.4E6, -0.3meg, -0.3MEG*
kilo	10^{3}	E3	k	KIL	*1000, 1e3, 1k, 1kil*
milli	10^{-3}	E-3	m	MIL	*0.0105, 1.05E-2, 10.5M, 10.5MIL*
micro	10^{-6}	E-6	u	MIC	*0.000005, 5e-6, 5u, 5mic*
nano	10^{-9}	E-9	n	NAN	*40E-9, 40n, 40nan*
pico	10^{-12}	E-12	p	PIC	*100E-12, 100P, 100PIC*
femto	10^{-15}	E-15	f	FEM	*9E-15, 9F, 9FEM*

The comma is reserved for separating parameters in lists. The period (dot) is reserved as a decimal point. "M" is interpreted as 10^{-3}, not as 10^{6}.

SI Units

All units used from simulator immanent components are derived from the SI Units system.

Quantity	Unit Name	Symbol
Length	Meters	m
Mass	Kilograms	kg
Time	Seconds	s
Electrical current intensity	Amperes	A
Temperature	Kelvins	K
Voltage (derived SI unit)	Volts	V

Qualifier of Parameters

Components are characterized by different physical quantities. A resistor, for example, is represented by current and voltage in the simulation. System variables may be accessed by reading (to use the actual quantity in expressions or to create an output) or writing (influencing quantities). Use the following syntax to obtain access:

```
Name.Qualifier
```

Computations and outputs require access to system variables. The form and number of the qualifier depend on the corresponding component.

All qualifiers are case sensitive and must use capital letters (that is R.V not ~~R.v~~).

Qualifier List

The following table represents the most common SIMPLORER qualifiers. The complete qualifiers are found with the corresponding model description.

Notation	Specification
Simulator Outputs	
`Name.V`	component voltage, node potential (read)
`Name.I`	component current (read)
`Name.dV`	derivative of the component voltage (read)
`Name.dI`	derivative of the component current (read)
`Name.VAL`	block/time function/characteristic output signal (read)
`Z_Name.ST`	yields the currently valid status of the state z_name true-1-marked-active false-0-unmarked-inactive
`Component Parameters`	
`Name.R`	component resistance (read/write)
`Name.C`	component capacitance (read/write)
`Name.L`	component inductance (read/write)

Notation	Specification
`Name.G`	component conductivity (read/write)
`Name.I0`	component initial voltage (read/write only at simulation start)
`Name.V0`	component initial current (read/write only at simulation start)
`Name.CTRL`	control signal (read/write)
`Name.UL`	upper limit (read/write)
`Name.LL`	lower limit (read/write)
`Name.FREQU`	frequency of a function (read/write)
`Name.TPERIO`	cycle duration of a function (read/write)
`Name.AMPL`	amplitude of a function (read/write)
`Name.INPUT`	block input signal (read/write)
`Name.TS`	sampling time of the block sampling function (read/write)
`Name.QUANT`	control quantity (read/write)
`Name.CH`	component characteristic (read/write)
`Name.FILE`	file name (read/write)

Parameter Types

The following table summarizes the parameter types used in Schematic dialogs.

Type	Value Utilization	Accepted Value Format
Common Type	All expressions are interpreted with their actual value to define a parameter.	All numerical values (constant with or without unit suffix), simulation parameters, variables, component parameters, mathematical, or logical expressions. A logical expression provides only the value '1' (TRUE) or '0' (FALSE). Examples: *10k*, 5K, *-1E-3*, *0.003*, 20MEG *Tend*, *Hmin*, *SECM.ITERAT* *var1*, *var2*, _var3, *var_4*, *Var_4* *R23.I*, C17.V, *E4.EMF*, *GZ1.VAL* *10*t+var1-INTEG(var2)* *delay>=2.5m*var1*
Control Type (Name.`CTRL`)	The expression value tests if >0, <0, or = 0. One exception is the VCO component. All values of this control parameter are used.	
Initial Value Type (Name.`x0`)	See common type. Initial values are set only once at simulation start.	
Logical Expression Type	Logical expressions are interpreted as TRUE if the provided value is '1'. Otherwise, the value is FALSE.	
Quantity Type (Name.`QUANT`)	All quantities are interpreted with their actual value to define a parameter.	The type accepts only voltage and current of voltmeters, ammeters, or wattmeters. The values cannot be assigned by a variable. Examples: *VM1.V*, *AM2.I*, *WM3.V*, *WM3.I*

Type	Value Utilization	Accepted Value Format
Characteristic Type (Name.CH)	For a given X value, the Y value is determined. The LOOKUP function provides characteristic values in equations. The value of Name.CH is irrelevant.	The type accepts only the name of a characteristic component. The values cannot be assigned by a variable. Examples: *EQUL1.VAL, XY1.VAL*
File Type (Name.FILE)	For a given X value, the Y value is selected. The value of Name.FILE is irrelevant.	The type accepts only a name referring to an .mdx file (.xls 4.0, .csv, .txt, .mdb). The value cannot be assigned by a variable. Example: *C:\release\diode.mdx*
Wizard (Dialog) Type	The values are used instantly to define dialog input settings, for example to provide input fields for coefficients of order dependent components.	Accepts only numerical values (constant with or without unit suffix). The values cannot be assigned by a variable. Examples: *10k*, 5K, *-1E-3*, *0.003*, 20MEG

Predefined Variables

The simulator uses predefined variables for internal computation. If these variables are used in a model description, an error message or unexpected effects result. All predefined variables are case insensitive. See also “Using Simulation Parameters” on page 53.

Predefined variables cannot be used for names in a model description.

System constants (read)	F, T, H, PI, TRUE, FALSE, SECM.ITERAT, FSTEP
General simulation parameters (read/write)	TEND, HMIN, HMAX, THETA, FSTART, FEND
SML keywords	MODELDEF, PORT, VAR, STORE, SIMCFG, OUTCFG, RESULT, SIMCTL, OUTCTL, RUN, INTERN, MODEL, UMODEL, COUPL, ELECTRICAL, MECHANICAL, REAL, INT, BIT, COMPLEX

Predefined Constants

The simulator provides natural and mathematical constants that can be used in mathematical expressions within component dialogs or SML descriptions. The following table shows the available constants and their corresponding symbols:

Symbol	Value	Unit	Description	Variable
π	3.141592654	[/]	PI	`MATH_PI`
E	2.718281828	[/]	Euler number	`MATH_E`
ε_0	8.85419 10^{-12}	$C^2 \bullet Jm$	Permittivity of vacuum	`PHYS_E0`
μ_0	1.25664 10^{-06}	T^2m^3/J	Permeability of vacuum	`PHYS_MU0`
k_B	1.38066 10^{-23}	J/K	Boltzmann constant	`PHYS_K`
e	1.60217733 10^{-19}	C	Elementary charge	`PHYS_Q`
c	299 792 458	m/s	Speed of light	`PHYS_C`
g	9.80665	m/s^2	Acceleration due to gravity	`PHYS_G`
h	6.6260755 10^{-34}	Js	Planck constant	`PHYS_H`
ϑ	-273.15	°C	Absolute Zero	`PHYS_T0`

Equations, Expressions, and Variables

Equations consist of operands and operators. An operand can be any number or variable name. An operator compares or assigns a value. In expressions you can create and use variables as often as you want. A variable is defined when the variable name is used in an expression or for a parameter value within a component dialog. You do not need to define the variable in a specific assignment unless you want it to have a defined initial value.

Z:=Y+X;   **X**,**Y**, and **Z** are the operands, and **:=** and **+** are the operators.

If operants are complex numbers (for example in an AC simulation), the comparison operators (<, >, <=, >=) consider only the real part.

Assignment operators	:=	Assignment
	##	Delay operator combined with the action type DEL
Arithmetic operators	*	Multiplication
	/	Division
	+	Addition
	–	Subtraction
Comparison operators without synchronization	<	Less than
	>	Greater than
	<> or ><	Not equal to
Comparison operators with synchronization	This operator type forces the simulator to synchronize on the condition with the minimum step width.	
	<= or =<	Less than or equal to
	>= or =>	Greater than or equal to
	=	Equal to

Logic operators (must be surrounded by spaces)	`AND`	Logical AND (conjunction)
	`OR`	Logical OR (disjunction)
	`NOT`	Logical NOT (negation)

Standard Mathematical Functions

Mathematical functions consist of the function name and one or two arguments. An argument can be any number or variable name. An mathematical function applies the function, which it represents, to the argument(s).

`r:=FCT(x,y),r:=FCT(z)`

x, **y, and z** are arguments, z is a complex number, FCT is the function name, **r** is the result.

If the argument(s) are complex numbers (for example in an AC simulation), the functions RAD, DEG, DEGEL, MOD, INT, FRAC, LOOKUP consider only the real part.

Notation	**Description**	**Example**
`SIN(x)`	Sine, x[rad]	SIN(PI/6)=0.5
`COS(x)`	Cosine, x[rad]	COS(2•PI/3)=-0.5
`TAN(x)`	Tangent, x[rad]	TAN(PI/4)=1
`ARCSIN(x)`; `ASIN(x)`	Arc sine [rad]	ASIN(0.5)=0.524=PI/6
`ARCCOS(x)`; `ACOS(x)`	Arc cosine [rad]	ACOS(0.5)=1.0471=PI/3
`ARCTAN(x)`; `ATAN(x)`	Arc tangent [rad]	ATAN(1)=0.785=PI/4
`ARCTAN2(x,y);` `ATAN2(x,y)` ATAN2=ATAN(y/x)	Arc tangent2 [rad] r=0 if x=0 and y=0; $-\pi \le r \le \pi$	ATAN2(.25,1)= ATAN(4)=1.325
`SINH(x)`	Sine hyperbola.	SINH(1)=1.175
`COSH(x)`	Cosine hyperbola.	COSH(1)=1.543
`TANH(x)`	Tangent hyperbola.	TANH(1)=0.762
`SQU(x)`	Square.	SQU(16)=16^2=256
`X^Y`	Power.	7^4=2401
`SQRT(x)`	Square root.	SQRT(9)=$\sqrt[2]{9}$=3
`ROOT(x,[y])`, y=2	n-th Root.	ROOT(27,3)=$\sqrt[3]{27}$=3
`EXP(x)`	Exponential function.	EXP(5)=e^5=148.41
`ABS(x)`	Absolute value.	ABS(-8.5)=|-8.5|=8.5
`LN(x)`	Natural logarithm.	LN(3)=$\log_e 3$=1.099
`LOG(x[,y])`; y=10	Common logarithm.	LOG(7,4)=$\log_4 7$=1.403
`INTEG(x)`	Integration of a variable from the function call until to the simulation end.	INTEG(var1)=∫var1 dx
`RE(z)`	Real part	RE(z)=5
`IM(z)`	Imaginary part	IM(z)=3
`ARG(z)`	Argument of a complex number in radians.	ARG(z)=0.53

3

Notation	Description	Example
	$z=a+bi=r(\cos\varphi+i\bullet\sin\varphi)=r\bullet e^{i\varphi}$ $z=5+3i=5.83(\cos30.96°+i\bullet\sin30.96°)$	
`SGN(x)`	Sign dependent value (-1, 0, 1). r=0 if z=0, 1 if Re(z)>0 or (Re(z)=0 and Im(z)>0), -1 otherwise.	SGN(3)=1; SIGN(0)=0; SIGN(-3)=-1
`RAD(x)`	Conversion from degrees to radians.	RAD(30)=PI/6=0.524
`DEG(x)`	Conversion from radians to degrees.	DEG(PI/2)=90°
`DEGEL(x[,y]); y=1`	Conversion from degrees electrical to seconds with respect to Hz.	DEGEL(180,50)=10ms
`MOD(x, [y]);` y=1	Modulus.	MOD(370,60)=10
`INT(x)`	Integer part of a value.	INT(2.5)=2
`FRAC(x)`	Fractional part of a value.	FRAC(2.5)=0.5
`LOOKUP(x,y)` x=Characteristic name y=X value	Access function to a characteristic.	LOOKUP(XY1.VAL,5)= Y value of the characteristic XY1 for the X value 5

`IF` (condition)
{ var:=1; }
[`ELSE IF` (condition)
{ var:=2; }
`ELSE`
{ var:=3; }]
	If-Else function to perform operations dependent on conditions.
	The ELSE IF and ELSE statement can be omitted.
		IF (t>=1)
		{ var:=1; }
		[ELSE IF (t>=2)
		{ var:=2; }
		ELSE
		{ var:=3; }

`DB(x)`	Conversion to Decibel (Available in the DAY Post Processor only.)	DB(8)=20•log(8)=18.062

When entering these functions, do not leave any spaces between the function arguments and the open parenthesis mark for example MOD(x).

To avoid potential errors during the simulation, if you define arguments for trigonometric functions, you must consider poles.

3.2 Schematic Operating Environment

Schematic is the heart of SIMPLORER's simulation tools. From Schematic, you can describe model structures, display simulation results with Display Elements, and launch other applications, such as an Optimetrics to perform a design optimization. The embedded Model Agent provides hierarchically arranged components. Drag-and-drop allows easy placement and connection of components to models. Schematic checks the validity of the input, and displays warnings and errors in the Information window. A variety of drawing tools allow you to illustrate and document simulation models.

<u>This section contains information on:</u>

- Schematic's main window
- Creating simulation models
- Simulating models
- Creating subsheets
- Editing model parameters

Double-click the Schematic symbol or choose PROGRAMS>SCHEMATIC to start the Schematic in the SSC Commander. If a project file is selected in the file list, double-clicking starts Schematic with this file. If the entry **{New File}** is selected, double-clicking starts Schematic with a new file.

Schematic Main Window

From SIMPLORER's Schematic, you can create simulation blocks, control the simulation process, and display simulation results. A variety of displays are available for data visualization.

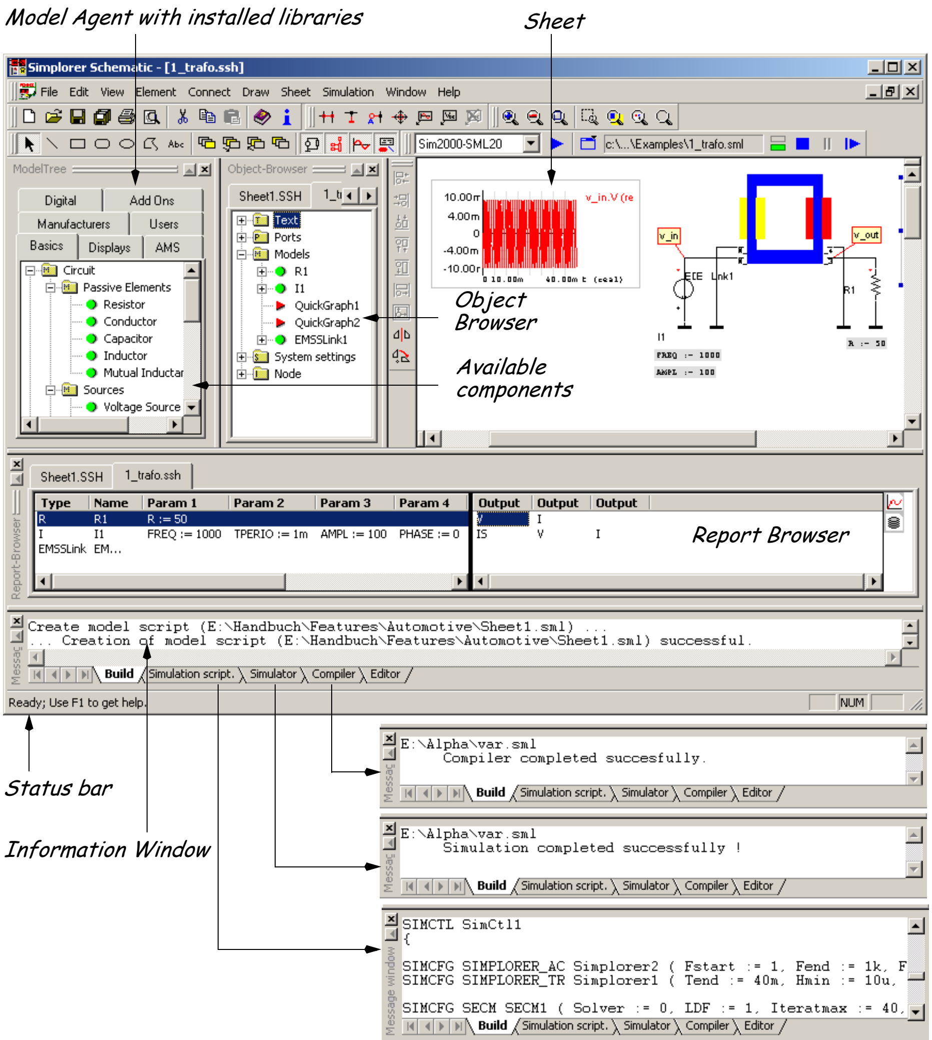
The Schematic main window consists of the following elements, which you can show and hide using the VIEW menu commands.

Model Agent: With installed libraries and components.

Object Browser: Shows all elements of a simulation model and their properties.

Information Window: Shows messages from Schematic (Build), the Simulator, the Compiler, and the embedded editor, as well as the Simulation model in SML (simulation script).

Report Browser: Lists all components of a simulation model and their parameters.



Schematic Sheet

A working sheet contains all data of a simulation model, including all component parameters and settings for the simulation. The Schematic files also contains displays, graphical elements, and printer settings.

Model Tree

The Model tree is based on the Model Agent, which is SIMPLORER's model database of all installed model libraries. Within Schematic, the Model tree is displayed in a separate window and shows all available components in tree structure. The functions are identical to the separate program Model Agent that can be started from the SSC Commander. See also 15 "Model Libraries" on page 479.

Object Browser

The Object Browser is displayed in a separate window and contains all components of a sheet and their properties. You can change element parameters directly in the Object Browser tree without opening an element's property dialog on the sheet.

You can also change the processing sequence for the simulation by selecting an element and dragging it to a new location in the browser tree. The sequence of processing is important for simulation models, where in one simulation step quantities use the calculated value of another one.

Information Window

When the information window is visible, warnings and error messages from the sheet, compiler and simulator are displayed. These messages help you find errors in the model description. In the Information Window a shortcut menu can be opened with a right mouse click in the window. The messages can be copied into the clipboard and inserted into any editor.

Report Browser

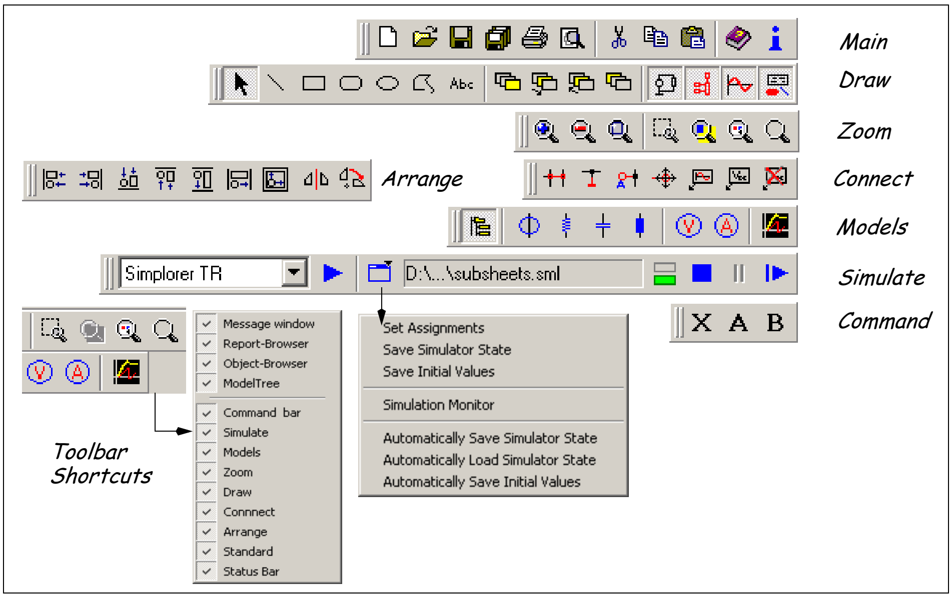
The Report Browser is displayed in a separate window just as the other parts of the Schematic. It contains all components of the simulation model and their parameters and outputs in list form. You can edit all names, parameters, and outputs in the list. The changes are immediately updated in the sheet.

Status Bar

The status bar at the bottom of the Schematic window shows information about selected items in the menu or toolbars. In addition, a red bar indicates the progress when an active file is saved manually or automatically.

Schematic Toolbars

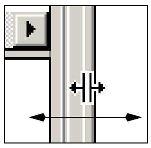
In Schematic, besides the Model Tree, the Object Browser, the Information Window, and the Report Browser, there are also several toolbars you can show or hide (VIEW>TOOLBARS). The following toolbars are available: Main, Models, Zoom, Draw, Connect, Arrange, Standard, and, if available, Command bar. Each of the toolbar commands can also be accessed from the Schematic menus.



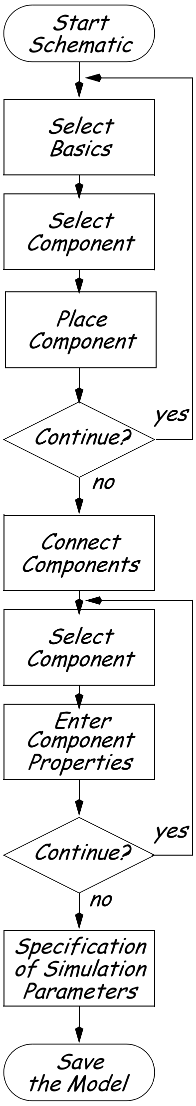
Changing the Arrangement of Schematic Elements

You can move a Schematic element within the window by clicking the border of the element and dragging it to the desired position.

You can change the size of Schematic elements by moving the sliders. When the mouse pointer is over a slider, it converts to a double line. By holding the left mouse button, you can move the slider to a new position.



3.3 Creating Simulation Models



- Click the desired tab in the Installed Libraries window.
- Select the component in the Available Components window.
- Drag the component with **pressed** mouse button onto the sheet (Drag-and-Drop).
- The selected component can be rotated (R key) or flipped (F key).

- Start the CONNECT>WIRE mode.
- Place the cross wires on the element pins.
- Set beginnings, corners, and ends of the wire with the mouse.
- Exit the wire mode with ESC key.

- Select component with the mouse.
- Open the shortcut menu with **right** mouse button.
- Enter parameters and outputs in «**Properties**».

- Click with the right mouse button in the sheet itself (not on any of the components).
- Enter simulation parameters in «**Simulation Parameters**».

- Save Simulation model:
  FILE>SAVE AS *name.ssh*

The terms component, model, and element are easily confused. In general, a simulation model, represented by a sheet, consists of components, elements, and simulation parameters.

Simulation Model	Components	Smallest parts of a complete simulation model (internal components, C models, macros).
	Elements	Displays, connections, name and parameter displays of components on the sheet, drawing elements, text elements.
	Simulation parameters	Simulation settings for the simulator and compiler.

3.3.1 Selecting Components

The Schematic Model tree provides all installed SIMPLORER libraries in the following tabs: «Basics», «Displays», «AMS», «Digital», «Tools», «Add Ons», «Manufacturers», «Users», and «Projects». Each tab can contain one or more libraries. All components are contained in a library.

Select a tab in the Installed Libraries window and then the desired component in the tree. Drag the component with pressed mouse button onto the sheet (Drag-and-Drop).

SIMPLORER Components

A simulation model (SML description) can consist of different model types. A model, used in a SML description, can be an internal component, an C model (modeled in C++), or a macro. A macro itself is a substructure that can include all model types: internal components, C models, and macros. There are two macro formats, graphical and text macros. VHDL-AMS models appear in the Model Agent as text macros and in the SML description as COUPL statement.

Model Type	**Internal components**	**C Models**	**Macros**		
Format	–	–	Graphical	Text	
Language	–	C++	SML	SML	VHDL-AMS
SML statement	INTERN	UMODEL	MODELDEF MODEL	MODEL	COUPL

Each model type has a corresponding symbol in the Model Agent. All components (internal components, C models, macros) belong to a library. They can be placed on the sheet and used in a simulation model in the same way. See also "Symbols in the Model Agent" on page 488.

Macros can be changed if the description is not encoded. All VHDL-AMS models are open coded and can be used to derive more advanced models.

Simulation Level — Static and Dynamic Components

For most applications, system level semiconductor components supply sufficient simulation data. However, if your simulation targets other data, such as turn on and off, losses, and thermal analysis, then you need to use dynamic components (Semiconductor device level components, Spice components, Manufacturer models). Keep in mind that using a large number of dynamic components increases simulation time.

It is not very useful to combine nonlinearity, an exact geometric representation, and the complete dynamic behavior in one model. As it is common in other areas of modelling too, only subsystems have to be described with high accuracy while other parts can be simplified.

Semiconductor Device Level and Manufacturer Components

Device level components appear in the «Basics» tab (Semiconductor Device Level, Spice) and «Manufacturers» tab. In SIMPLORER, there are two types of manufacturer models:

- Parametrized models, in which parameters can be changed («Basics»).
- Completely parametrized models, in which parameters can**not** be changed («Manufacturers»).

Component Overview

SIMPLORER includes many standard components stored in various libraries. When you are using SIMPLORER for the first time, it may be difficult to locate all the available components. The following list should help you to find the components you want in the default libraries:

Basics Tab

The **«Basics»** tab provides electric circuit components, blocks, states, measuring devices, signal characteristics (functions to evaluate characteristics online during simulation), modeling tools (time functions, characteristics, equations), and components of physical domains.

- Circuit Components on page 71
- Block Diagram on page 201
- State Graph on page 241
- Measurement on page 249
- Signal Characteristics on page 255
- Tools on page 275
- Physical Domains on page 305

Displays Tab

The **«Display»** tab provides elements for visual online display of graphical and numerical simulation outputs during a simulation on the Schematic sheet.

- Displays on page 369
- Displays connected on page 369
- Multimedia on page 382
- Data Channels on page 382
- Signal Processing on page 381

AMS Tab

The **«AMS»** tab provides electric circuit components, blocks, measuring devices, modeling tools (Time functions), and components of several physical domains modeled in VHDL-AMS. You find the model descriptions in the SMPLORER online help.

- Circuit Components
- Block Diagram
- Measurement
- Tools
- Physical Domains

Digital Tab

The **«Digital»** tab provides VHDL-AMS components with common basic functionality used for simple digital circuits . You find the model descriptions in the SMPLORER online help.

- Digital Sources
- Counters
- Flip-Flops
- Latches
- Logic Blocks
- Logic Gates

Add Ons Tab

The **«Add Ons»** tab provides advanced model libraries for different application fields. You find the model descriptions in the SMPLORER online help. The Compatibility library is required to use simulation sheets of former versions.

Advanced Libraries:

- Electric power components (power6.smd)*
- Set of components suited for needs of the automotive industry (automotive6.smd)*
- One-Dimensional Mechanical System Module (mechsim6.smd)*
- Hydraulic components (hydraulic6.smd)*
- Library with machine models (*_ist6.smd)*
- Nonlinear transmission and linear transmission components (transfer6.smd)

Interfaces:

- ECE (equivalent circuit extraction) Maxwell interface (interface.smd)
- RMxprt interface (interface.smd)
- Full-Wave SPICE interface (interface.smd)
- Mathcad interface (interface.smd)
- Matlab/Simulink interface (interface.smd)*

Compatibility:

- SIMPLORER 4: Semiconductors device level of former SIMPLORER versions
- Digital: Digital Basic Gates in different modeling levels (NAND, NOR, …).
- Magnetic: Elements for nonlinear magnetic circuit modeling.
- Function Blocks: Functional blocks libraries (Integrator, Frequency Doubler…).
- Lines: Macros, based on the gamma equivalent circuit, to simulate line segments.

*These libraries are optional (not included in the basic SIMPLORER package).

Tools Tab

The **«Tools»** tab provides components to calculate coordinate transformations, components to connect conservative nodes from different natures, and components to connect different data types. You find the model descriptions in the SMPLORER online help.

Manufacturers Tab

The **«Manufactures»** tab provides libraries with semiconductor device level models of different manufactures. You find the model descriptions in the SMPLORER online help.

Analog Devices, Fuij, Maxim, International Rectifier, Siemens, Semicron… and other manufacturer libraries with pre-parameterized components.

Users Tab

The **«User»** tab provides a location to insert user-defined libraries. You can insert your own libraries also in all other tabs.

Projects Tab

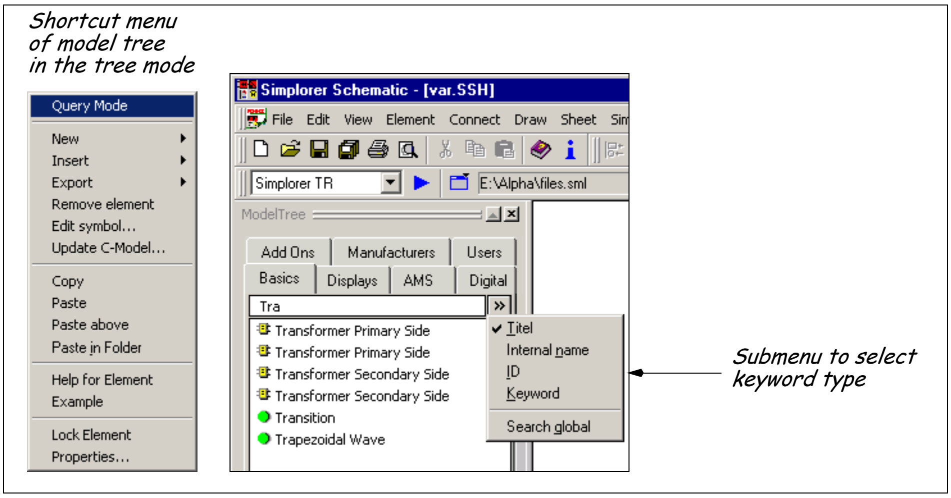
The **«Project»** tab provides libraries included in an opened project. This tab is only available if you work with a project. When you use the PROJECT>COPY command in the SSC Commander, libraries in these tab are considered as project files and added to the copied project.

Searching for Components

Searching for components in the installed libraries can be time-consuming. The Search feature minimizes the complexity of the search and allows quick access to all models.

The Search feature is available only in Query mode. You can change modes using the shortcut menu (click with the right mouse button in the model tree).

In the search string field, the  button accesses the submenu for the keywords used in the query mode.



You can search the following fields:

Title: Specification in the model tree.

Internal Name: The name of the model, for example CNL, EI

ID: Internal ID Number of the model.

Keyword: Keyword, which identifies a model; a model can have several keywords.

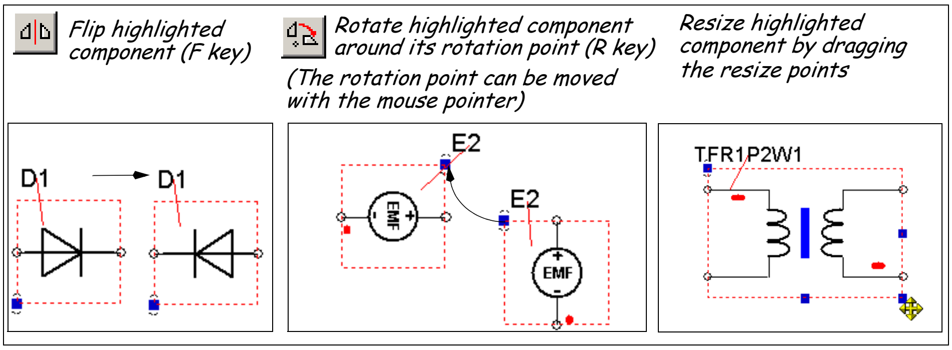
After the keyword is entered in the input field, the display underneath are updated immediately. The input is case sensitive. You can only look for keywords which are entered for the models. New created language resources or user-defined models have no entries for keywords. In this case you can only look for titles and internal names.

The search can be carried out within the selected library or in all installed libraries («**Search global**»).

Arranging Components

Each selected component placed on the sheet can be moved with drag-and-drop, rotated with the R or flipped with the F key. A selecting component is bordered by a a red broken line. The F and R key are available for all selected components on a sheet.

In addition, you can change the symbol size by dragging on the blue sizing handles with the mouse pointer. The size is adapt to the grid which is used for all symbols on a Schematic sheet.



If a symbol has the resizable option you can adapt the symbol size on the sheet. See also “Defining Sheet and Symbol Properties” on page 503.

3.3.2 Using Pins

Pins are graphical connectors for linking nodes or parameters with a wire. There are two different pin types:

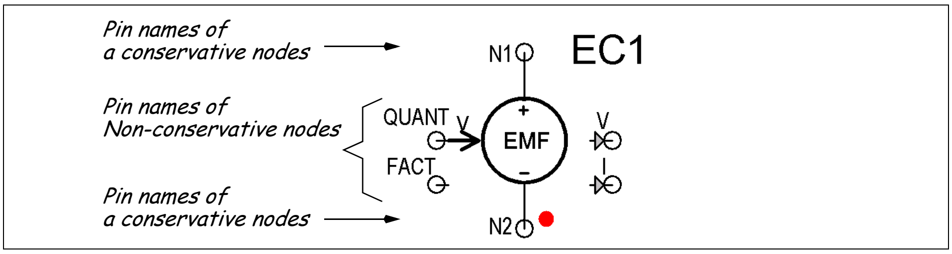
- Conservative Nodes
- Non-conservative Nodes, including Parameters, Inputs, and Outputs

Conservative nodes have a nature type. You can connect only nodes with the same nature type directly. To connect different natures use the Domain-to-Domain component.

Non-conservative nodes represent different data types.You can connect only nodes with the same data type directly. To connect different data types Omnicaster components must be used. Non-conservative nodes can be displayed or hidden. If you hide a component pin, you can define the connection through a name reference.

Conservative and Non-Conservative Nodes

Conservative nodes belong to components of physical domains (electrical, fluidic, magnetic, mechanical, thermal). If you want to create a topology, you must not hide a conservative node pin because doing so would make it impossible to define a connection. Conservative nodes have no direction attribute.



Non-conservative nodes behave differently. They belong to circuit as well as block diagram and state graph components. Non-conservative node pins can be hidden, since a connection can be produced by entering a name. Non-conservative nodes have one of the following direction attributes: IN, OUT, or IN/OUT.

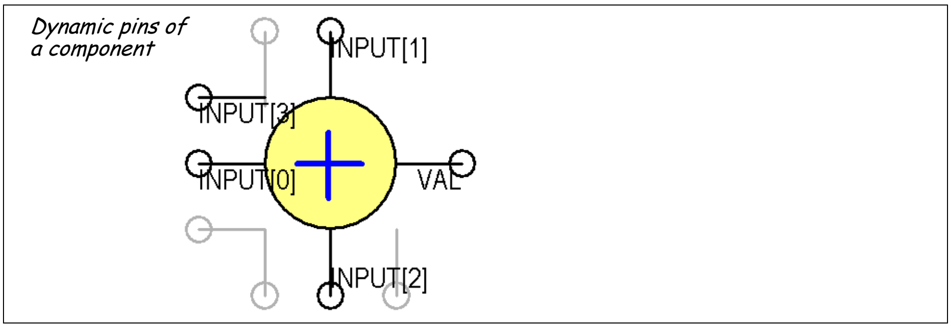
Displaying and Hiding Pins

You can display and hide each pin within the component property dialog. Double-click the symbol to open the property dialog and choose **«Output/Display»**. Then select the **«Pin»** box of each parameter which should be displayed on the sheet.

To select more than one entry in the list press the CTRL or SHIFT key and click all entries you want. Then click with right mouse button in the list and select **«Pin»**. Click <Apply> and all selected parameters and nodes get a pin on the sheet.

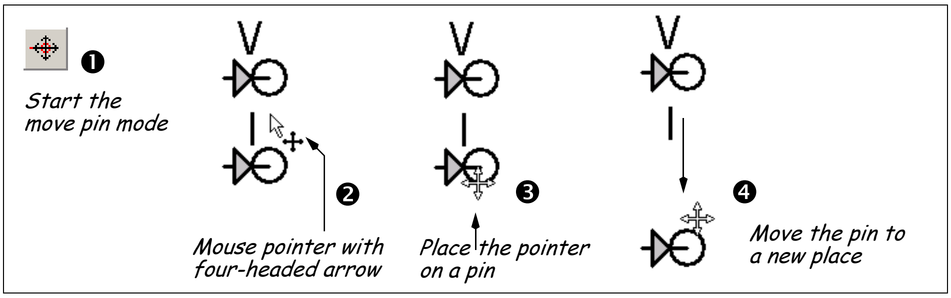
Dynamic Pins and Parameters

Some components as Summation, Multiplier, States, or Equations have no fixed number of inputs or parameters. If you create new pins within the property dialog Schematic arranges these pins around the symbol on the sheet. You can recognize dynamic parameters (pins) by the number in brackets in each name.



Moving Pins

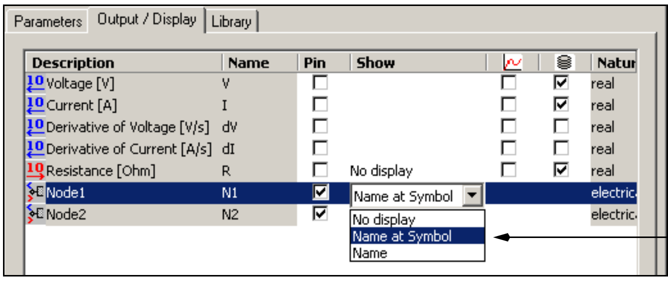
You can move all pins displayed on the sheet within the predefined grid. Start the moving pin mode with CONNECT>MOVE PIN or with the symbol on the toolbar. The mouse-pointer changes into a pointer with a four-headed arrow. Place the pointer on the component pin. If the pointer changes into a four-headed arrow drag the pin with pressed mouse button to the new place. Exit the move pin mode with <ESC>.



Names at Pins

For all nodes (conservative and non-conservative), the node name can be displayed on the sheet with two settings: **«At the symbol»** or **«Name»**. Double-click the symbol to open the property dialog and choose **«Output/Display»**. Then open the list box of the parameter from which the name should be displayed and select an entry.

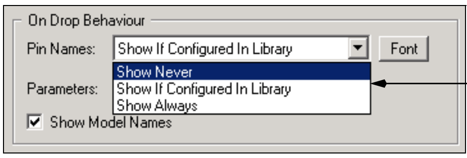
To select more than one item in the list press the CTRL or SHIFT key. Then click with the right mouse button in the list and select **«Show»** and your choice. Click <Apply> and all selected parameters and nodes are updated on the sheet.



Select a display option from the list

When you place a component, you can display or hide all pin names simultaneously using the SHEET>PROPERTIES command. Click the **«Sheet»** tab, and select «Show Never» (no display) or **«Show Always»** (all pin names are displayed) in the **«Pin Names»** list. The selection «Show as Configured in library» uses the definitions in the model library whether a pin name is displayed or not.The selection corresponds to the **«At the symbol»** function. Please note: The selections do not affect components that were already placed on the sheet.

In addition, you can also define the font for all pin names. However, the font is used also for placed components.



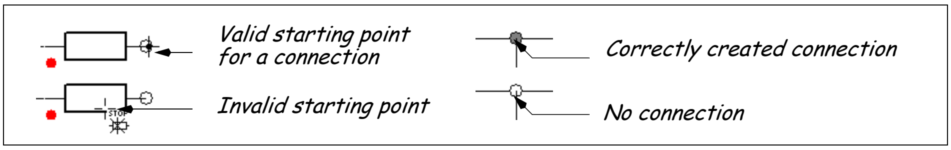
Select one of the options

3.3.3 Creating Connections

Connections between components are created in the Wire mode. Start this mode in one of the following ways:

- CTRL+W
- CONNECT>WIRE Menu
- sheet shortcut menu **«Wire»**
-  symbol on the toolbar

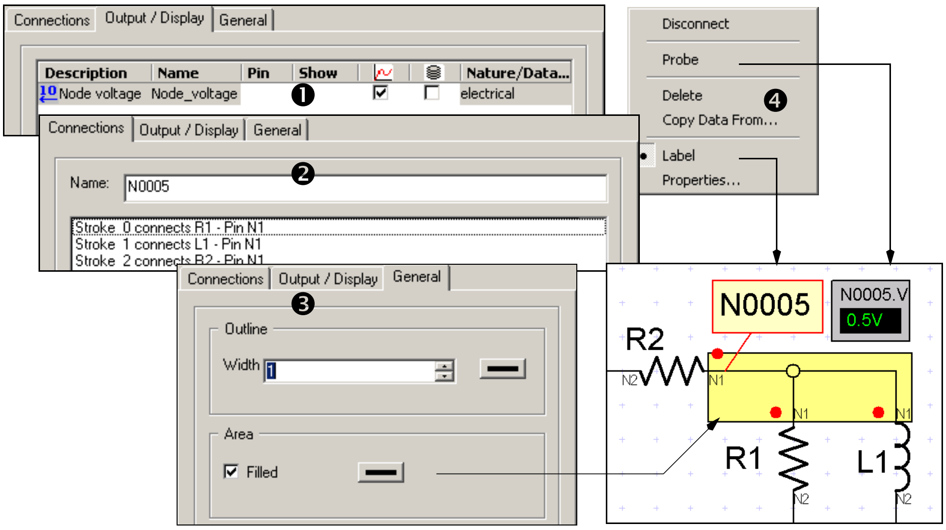
Place the cross wire on a component pin on the sheet, and set the beginning, corners, and end of the wire with the mouse. Press ESC to exit the wire mode.



Schematic checks the created model after the simulator starts. If connections are missing, an error message or warning is displayed in the «**Build**» area of the Info Window. Double-click the warning or error text to move the element with the wrong connection into the visible range of the Schematic.

Properties of Connections

Connections are elements with modifiable graphic properties. Double-click a connection or the choose EDIT>PROPERTIES to open the property dialog. In the «**Connections**» dialog the name of the connection and the connected components are displayed. The «**General**» dialog contains the graphic settings. See also "Displaying Node Names" on page 347.

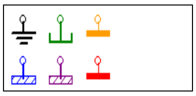
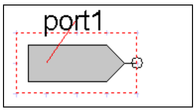
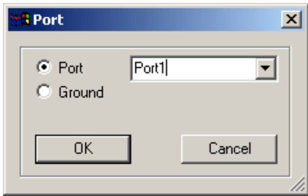
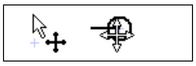
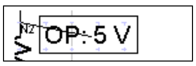


❶ *Dialog for defining of screen and file outputs*
❷ *Display of the components connected with the node*
❸ *Settings for the graphic representation*
❹ *Shortcut menu of a connection*

Connect Toolbar and Menu

The Connect toolbar and CONNECT menu offer further functions for connections.

Connect Menu

Command	Toolbar Symbol	Cursor Symbol	Description
WIRE			Starts the wire mode to connect components.
GROUND			Places a ground node on the sheet. Depending on the domain of the connected component the symbol changes shape and color.
PORT			Places a new port for external connections. Enter the port name in the input field and click <OK>. 
MOVE PIN			Starts the moving pin mode.
PROBE			Places a probe at the selected wire on the sheet. See also "Probe Elements" on page 380.
SHOW DC RESULTS			Displays results of an DC simulation at all corresponding components on the sheet.
HIDE DC RESULTS			Hides results of an DC simulation at all corresponding components on the sheet.

3.3.4 Defining Component Properties

Every component placed on the sheet has a property dialog to modify the component parameters. The form varies depending on the corresponding component.

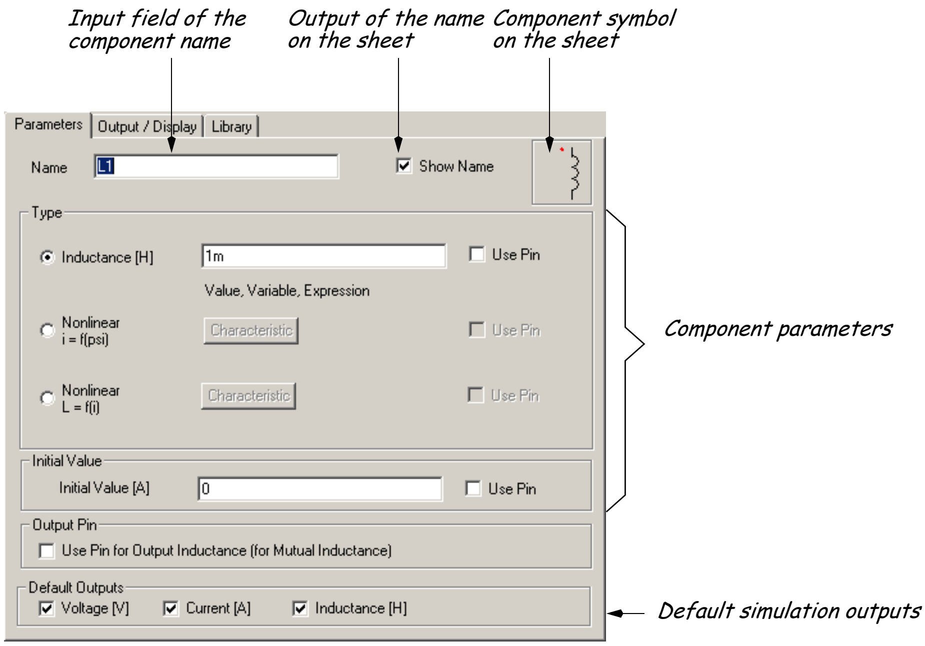
Do one of the following to open the property dialog:

- Double-click the component.
- Click with **right** mouse button the component, and choose **«Properties»**.
- Choose the EDIT>PROPERTIES command for the selected component.

Defining Parameters

Fixed Component Parameters

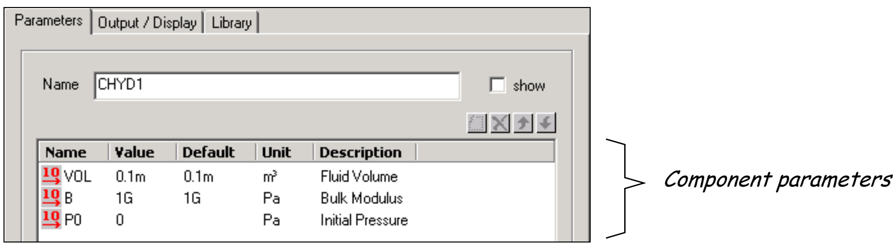
To change names and parameters, click in the desired text box. The cursor appears, and the text can be changed.



Press the TAB or the ENTER key to switch between the fields. If you click <Apply>, the changes are visible immediately on the sheet, and the **«Output/Display»** menu is updated without forcing you to close the dialog box. Otherwise, click <OK> to save the changes.

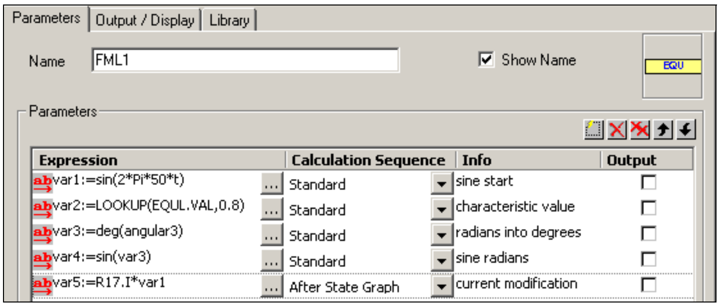
Components without special wizard dialogs (physical domain models, VHDL-AMS models, user-defined macros) use a dialog with a simple listing of the used parameters. In the list you can only modify the value of a parameter. Entries in the **«Name»**, **«Default»**, **«Unit»**, and **«Description»** column cannot be changed.

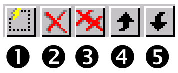
To define the value of a parameter you can also use a pin. Click the **«Output/Display»** tab and check the Pin box. If you use the pin for a parameter, you cannot change the entry in the **«Value»** field.



User-Defined Component Parameters

Some components have user-defined parameters; the number of parameters depends on the specifications in the component dialog. With the symbols on the upper right side, parameters can be created, deleted, or moved (changing the sequence in the list).





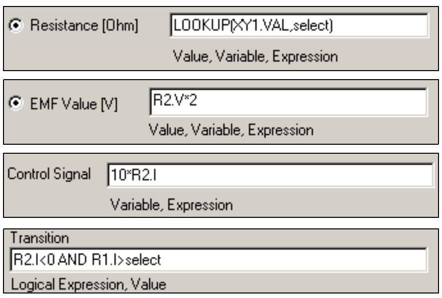
❶ *Create a new parameter*
❷ *Delete selected Parameter*
❸ *Delete all parameters in a list*
❹ *Move a selected entry up one position*
❺ *Move a selected entry down one position*

Using Parameter Names

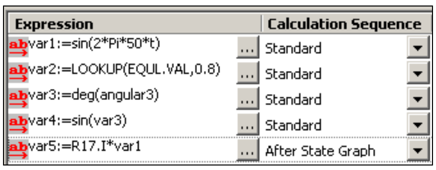
All component parameters can be used in expressions defining other component parameters or on the right-hand side of equations.

You access parameters using the following: `Name.Qualifier`. The “Name” is the name of the component. If you do not know the qualifier of a component parameter, look at the **«Output/Display»** page where the parameters and their qualifiers are listed. See also “Qualifier of Parameters” on page 29.

Parameter Input Fields:

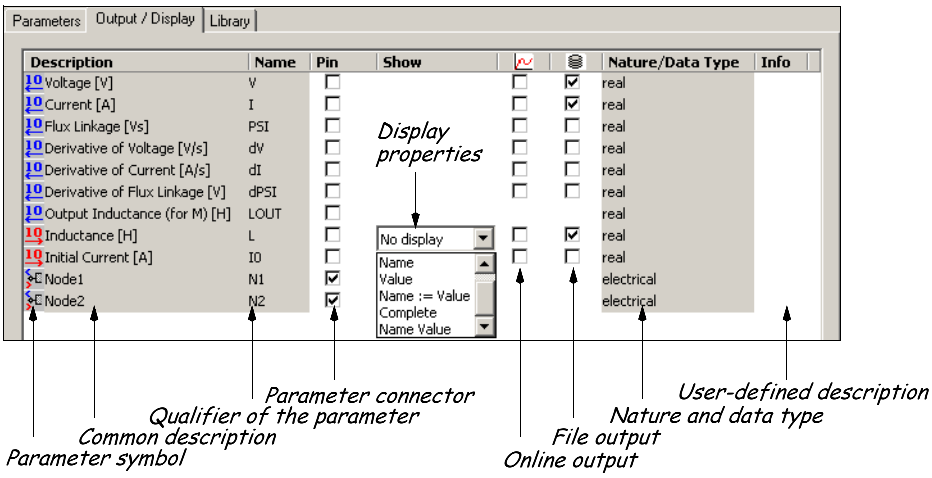


Equations:



Defining Displays and Outputs

Every component has an **«Output/Display»** dialog, as shown in the figure below. The number and type of the entries in the table depends on the component.



Explanations for Quantities - The Description Column

The entries in this column show the common parameter description of both the component parameters and simulation outputs. The symbol at the beginning characterize the parameter function. If the component has conservative nodes, they are also listed in the table. You cannot modify the entries in this column.

Value (in/out/bidirect.)	*Logical Expression (in/out/bidirect.)*
Initial Value (in/out/bidirect.)	*String Variable (in/out/bidirect.)*
Control Signal (in/out/bidirect.)	*Characteristic (in/out/bidirect.)*
Quantity Value (in/out/bidirect.)	*Conservative Node / File*
Vector(in/out/bidirect.)	

Qualifier of Parameters - The Name Column

The entries in this column show the qualifier of both the component parameters and simulation outputs. If the element has conservative nodes, they are also listed in the table. You need the qualifiers of parameters for using in equations and expressions. You cannot modify the entries in this column. See also "Qualifier of Parameters" on page 29.

Connectors for Quantities - The Pin Column

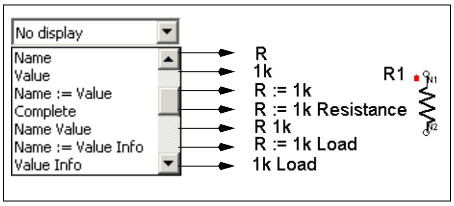
If the box of an entry is checked, the quantity is available as pin at the symbol. You can connect the quantity with another pin of a component to define the value. Please note: The pins have direction attributes. Component parameters represent input pins, simulation outputs outputs pins. To select more than one entry in the list press the CTRL or SHIFT key and click all entries you want. Then click with right mouse button in the list and select **«Pin»**. See also "Using Pins" on page 44.

Parameter Displays on the Sheet - The Show Column

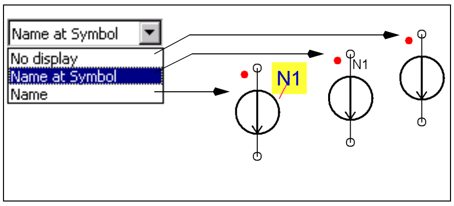
The figure below shows all the available choices of displays for components. If the **«Show»** column is clicked with the right mouse button, the drop down menu to select the display form of component parameters on the sheet is opened. The definition in this menu is valid only for the selected parameter. Please note: The display settings are available only for component parameters, not for simulator outputs.

To select more than one entry in the list press the CTRL or SHIFT key and click all entries you want. Then click with right mouse button in the list and select **«Show»**. See also 11.2.2 "Displaying Names and Parameters of Components and Connections" on page 347.

For Parameters



For Pins



Simulation Outputs for a Component - The Output Columns

If you select one or all of these boxes you define the corresponding output of a parameter. You can define an analog online output for Display Elements or the View Tool, and an offline output (.sdb file) for the DAY Post Processor or external programs. See also 12.1 "Representing Simulation Results" on page 365.

Properties of Nodes - The Nature/Data Type Column

The entries in this column show the data type of both the component parameters and simulation outputs. If the element has conservative nodes, the column shows the nature of them. You cannot modify the entries in this column.

Additional Information - The Info Column

The text lines in the **«Info»** column are modifiable of the user. The text is displayed when you select **«Complete»** as display option for an entry.

Library Tab

The **«Library»** tab shows the model library allocation and the model type identification. You cannot change these settings.

In addition, if the model has more than one modelling description, you can define the used descrition for simulation, SML or VHDL-AMS. For VHDL-AMS models, you can select the used architecture. See also "Creating Models in VHDL-AMS in the Model Agent" on page 492.

3.3.5 Using Additional Modeling Functions

In a simulation model, you can use components specified with parameters and included in a wiring topology to describe your simulation problem. Standard simulator functions and component parameters are available for defining parameters.

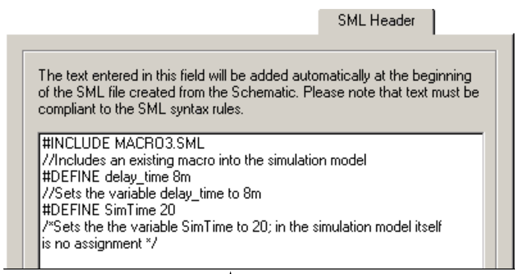
Schematic also offers additional functions to influence a model description, for example using simulation parameters, using SML text, defining initial values, using simulation state, and deactivating components.

Using Simulation Parameters

The following table lists the simulation parameters, which control the simulation, and how to access them. All simulation parameter names are case insensitive, but the qualifier SECM is case sensitive.To display, for example, the current time step you have to define an expression *var1:=h* and define *var1* as output.

System constants (read only)	`T, t`	Current simulation time in seconds.
	`H, h`	Current simulation time step in seconds.
	`PI`	System variable. π = 3.141592654
	`TRUE`	Logic operator. Provides the value '1'.
	`FALSE`	Logic operator. Provides the value '0'.
	`SECM.ITERAT`	Current number of iterations.
General simulation parameters (read/write)	`TEND`	Simulation end time in seconds (TR).
	`HMIN`	Minimum time step in seconds (TR).
	`HMAX`	Maximum time step in seconds (TR).
	`THETA`	Global ambient temperature for temperature dependent components (TR, AC, DC).
	`FSTART`	Start frequency of simulation in Hz (AC).
	`FEND`	End frequency of simulation in Hz (AC).
	`FSTEP`	Frequency step time of simulation in Hz (AC).
	`ACSWEEPTYPE`	Sweep type. 0=linear, 1=decadic (AC).
	`EMAX`	Maximum error in A (AC, DC).
	`ITERATMAX`	Maximum number of iterations (DC).
	`RELAXMAX`	Maximum number of relaxation (DC).
Circuit simulator (read/write)	`SECM.SOLVER`	Used integration method. 0=Euler, 1=Trapez
	`SECM.LDF`	Local discretization error.
	`SECM.ITERATMAX`	Maximum number of iterations for one simulation step.
	`SECM.IEMAX`	Maximum current sum error.
	`SECM.VEMAX`	Maximum voltage error.

SML Text in the Schematic



Edit window to create a SML header

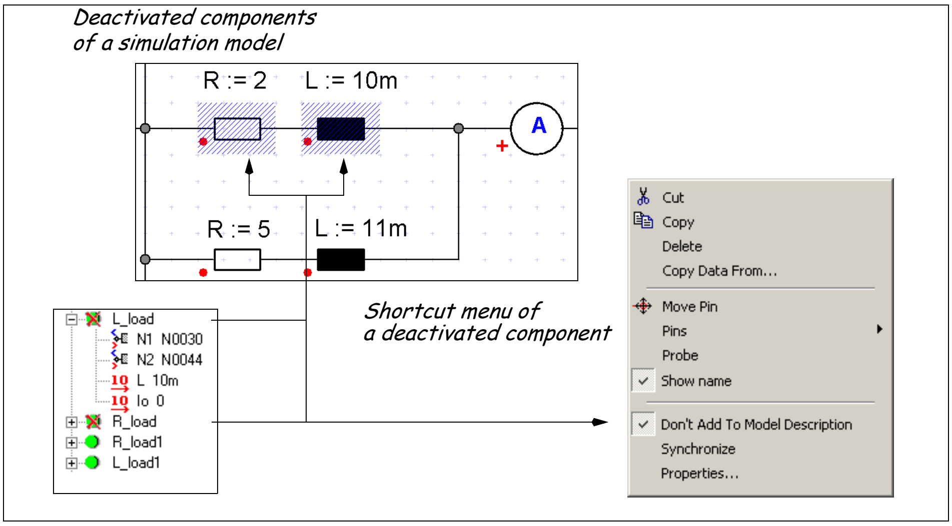
In the **«SML-Header»** dialog, statements in SML notation can be entered (for example #include). This header is inserted before the model's simulation script. You can define the header for the active model only or as a default (select **«Use as default»**).

Initial Values and Simulation State

When starting a simulation, you can use initial values and state files to assign special values for components in a simulation model. The simulator loads these files at the simulation start. To define an initial value file (.aws) choose SHEET>SIMULATION PARAMETERS, check the option **«Use Initial Values»**, and define an .aws file. See also 13.4.2 “Using Simulator Files” on page 412.

Deactivating Comment

Using the Deactivating Comment command, you can deactivate separate components or parts of a model sheet for a simulation run. The components and all their properties remain on the sheet, but the simulator ignores the deactivated components. The connection between the terminals is considered open. This function is especially helpful when simulation models must be tested with several elements and parameters. The Deactivate function is set with ELEMENT>DON'T ADD TO MODEL DESCRIPTION or the same command in the shortcut menu for the selected component. Deactivated components are hatched; the hatch color can be modified in the **«Sheet» dialog** of the model sheet (SHEET>PROPERTIES).



3.4 Characteristics in Simulation Models

Some components use characteristics for specifying of their model properties, for example nonlinear passive components, nonlinear sources, semiconductors of system level, the nonlinear transfer block, and the nonlinear DC machine.

SIMPLORER offers two types of characteristics: Predefined and user-defined. The characteristic components are located in the *Tools/Characteristic* folder in the **«Basics»** tab.

You can assign characteristics to a component in two different ways: Via the component dialog itself or via a separate characteristic component. The data can be stored in a separate .mdx, .xls, .csv, or .txt file.

The Sheet Scan Tool scans a printed characteristic curve and transfers it in a simulator legible format. See also “Separate Component Characteristic” on page 297.

Characteristics within the Component Dialog

The separate component characteristic dialog is used if you define a characteristic directly in the dialog of components using nonlinear characteristics. Click <Characteristic> to open the dialog.

Choose the option **«Lookup Table»** and enter the data-pairs in the table on the right side. Create, delete, or move the entries with the symbols placed on the upper right side.

Characteristics defined only in the lookup table create an .mdx file stored at the same position as the model file. You can use the automatically created file within the DAY Post Processor or as characteristic for other components. Please note: The file is overwritten with the active data in the lookup table at each simulation run.

Characteristic Reference via Separate Component

The characteristic via a name reference is used if you open the <Characteristic> dialog in a component using nonlinear characteristics and choose the option **«Reference»**. Type the name, for example XY1.VAL in the text box.

You can also connect the characteristic to the pin. Choose the nonlinear component type in a dialog and select the option **«Use Pin»**. The pin to connect the characteristic is available on the sheet.

Characteristics in Files

To use a file for defining a component characteristic you have to define the file name which refers to a proper characteristic file format. This input field is active when you had checked **«File Name»** in the <Characteristic> dialog in a component using nonlinear characteristics.

A two-dimensional data file, which first line is occupied with the axes names, is required. The XY-data file has always two channels (columns) where X-value and Y-value are stored. The length (number of lines) of the data files is arbitrary and limited only by the available memory. The symbols starts the dialog to open a data-pairs file or to save a file. Please note: Select the correct file filter (.mdx, .xls, .txt) before browsing. The characteristic file can be created by the DAY Post Processor or a text editor. See also 14.2 “SIMPLORER ASCII System Data Formats” on page 467.

SIMPLORER Sheet Scan Tool

Sheet Scan is a tool to extract characteristics from graphics (data sheets) and change the characteristics into the SIMPLORER data format. Start the program from the Windows taskbar: Start>Programs>SSC60>Tools>Sheet Scan.

Creating Characteristics and Functions with Sheet Scan

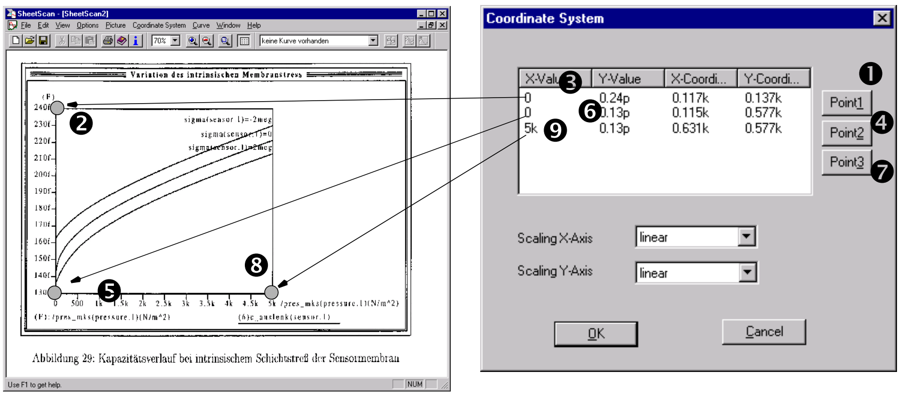
1 Scan the graphic with the required characteristics, and save the data in one of the following formats: .bmp, .jpg, .png, or .gif.

2 To start Sheet Scan from the Windows taskbar, click Start>Programs>SSC60>Tools>Sheet Scan.

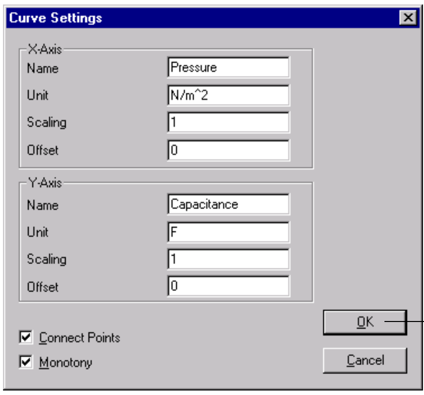
3 Choose FILE>NEW to create a new data sheet.

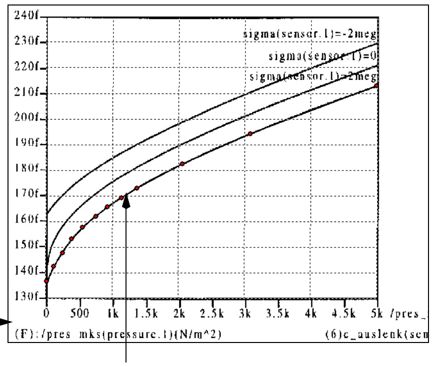
4 Choose PICTURE>LOAD PICTURE to load the graphic.

5 Define the coordinate system (COORDINATE SYSTEM>NEW) by clicking the corners of the coordinate system and entering the axis values for those points. Click <OK>. The grid is placed in the graphic.



6 Choose CURVE>NEW. In the displayed dialog define the properties of the curve and click <OK>. The cursor changes to cross hairs. Click the points of the characteristic which you want to apply. The points are connected automatically. A click with the right mouse button finishes the point selection mode.





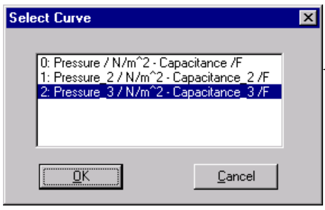
Define the points with the mouse

The shortcut menu contains various curve-editing functions (Insert, Delete, Append points).

7 To save the curve in SIMPLORER format, choose FILE>EXPORT and select .mdx file format.

If you want to save the graphic and the sampled curve, choose FILE>SAVE AS, and save the Sheet Scan .ssf file.

To create more than one curve in a graphic, choose CURVE>NEW. Existing curves move to the background. The CURVE>SELECT option brings the corresponding curve into the foreground, so you can edit it.



Dialog Curve Select

3.5 Subsheets in Simulation Models

Large schematics may make it difficult to see your work clearly because the screen area is limited.To use less screen space, subsheets, which allow you to structure the wiring and create hierarchical models, can be integrated into the mainsheet and nested as much as you want.

If block diagrams are part of a subsheet, the block sequence is considered only within the single subsheet; therefore, recursive operations with block diagrams could cause step delays during simulation.

Descriptions created from a subsheet definition can be used to add macros to model libraries in the Model Agent. Macros may be used in any model description, similar to each basic component (drag-and-drop from the Model Agent).

Schematic offers two ways to create subsheets, graphical and as text. Graphical subsheets are considered as SML models whereas text subsheets can be SML and VHDL-AMS models.

Subsheet symbols, used of the Schematic, can be modified.Choose **«Edit Symbol»** on the shortcut menu to start the Symbol Editor and change the symbol.

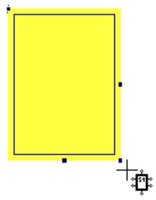
3.5.1 Graphical Subsheets

Graphical subsheets can be created as new definition or from a selection on the sheet. To create a new subsheet choose SHEET>SUBSHEET>NEW GRAPHICAL. To create a subsheet from a selection select the components and choose SHEET>SUBSHEET>NEW GRAPHICAL FROM SELECTION.

Creating New Subsheet

1 Choose SHEET>SUBSHEET>NEW GRAPHICAL. The mouse pointer becomes a cross wire.

2 Draw the subsheet symbol to the size desired by holding the left mouse button and then release the mouse button.

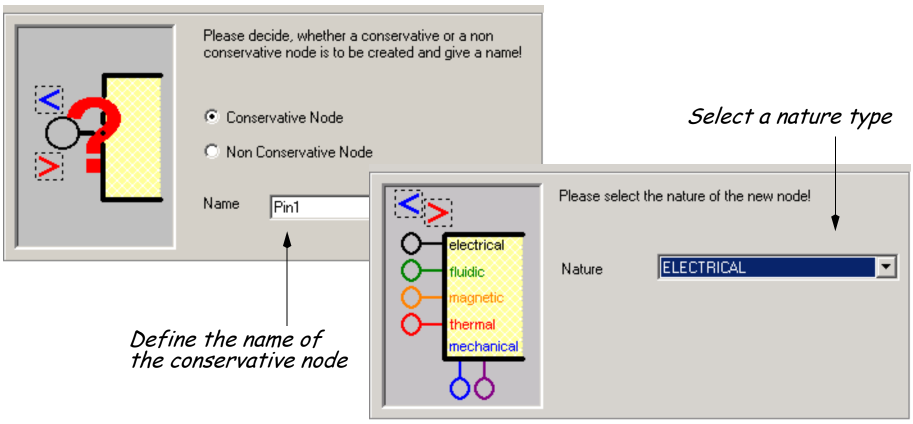


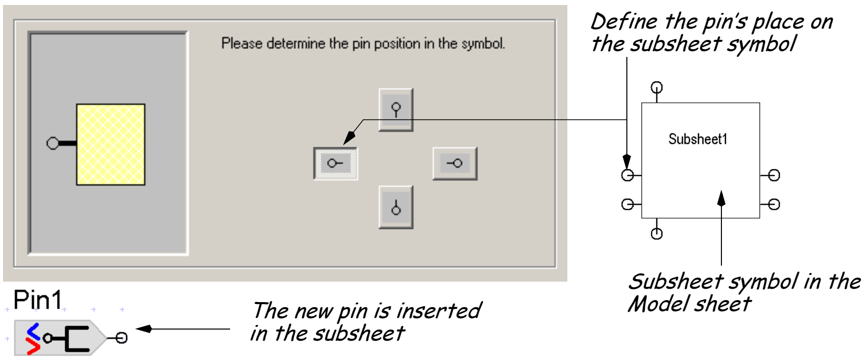
A rectangle for the macro is displayed and a new empty sheet is opened.

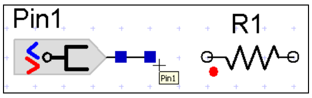
3 Create the subsheet model structure by placing components and wiring them.

4 Establish conservative nodes in SHEET>SUBSHEET>CREATE PIN/PARAMETER. Conservative nodes are connectors from the substructure to the next higher model level.

5 Define the name of the conservative node, click <OK> and select a nature type.





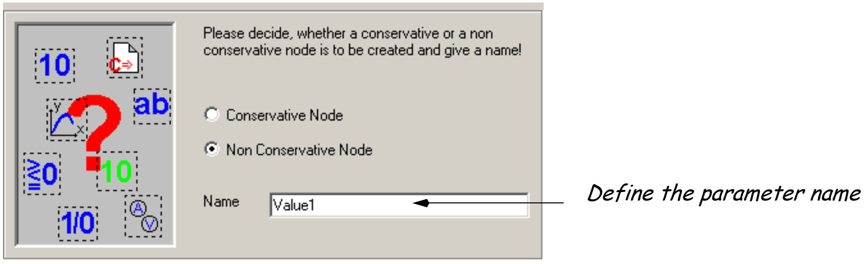


5 The new pin symbol can now be connected with the wiring.

6 Establish non-conservative nodes in SHEET>SUBSHEET>CREATE PIN/PARAMETER.

Non-conservative nodes are used to change parameters and to built graphical connections in block diagrams and state graphs. Non-conservative nodes need the additional information whether the pin should be used as input, output, or input and output.

Please note: There is a pin symbol for each non-conservative node, also when the pin is not connected with a model parameter.



7 Switch to the next higher model level with **«Level Up»** in the subsheet shortcut menu.

8 Double-click the subsheet symbol to open the property dialog and to define parameters.

The subsheet is now created and is regarded as a normal component in the system. To edit the subsheet, select the symbol and choose **«Open»** on the shortcut menu or choose SUBSHEET>OPEN.

Creating From Selection

When existing components on the sheet should be summarized as subsheet, you can select the components and choose SHEET>SUBSHEET>NEW GRAPHICAL FROM SELECTION. This function transfers all selected component into the subsheet.Conservative nodes are automatically created and connected.

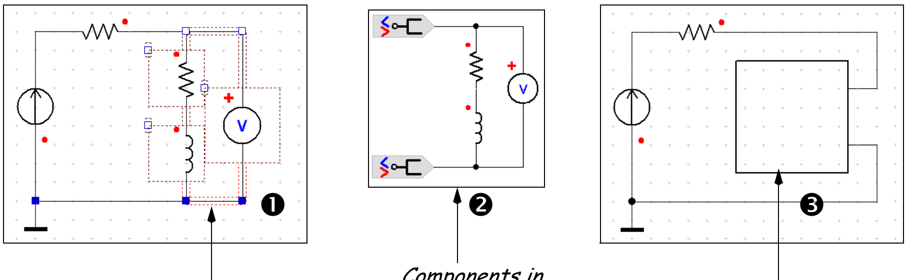
1 Define the model structure you want in the main sheet.

2 Mark all components which should be part of the subsheet.

3 Choose SHEET>SUBSHEET>NEW GRAPHICAL FROM SELECTION.

4 The subsheet with the selected components and the pins are created and opened. Define more parameters for inputs and outputs if required.

5 Change to the main sheet (**«Level Up»** in the subsheet shortcut menu) and check the connections and parameters of the subsheet.



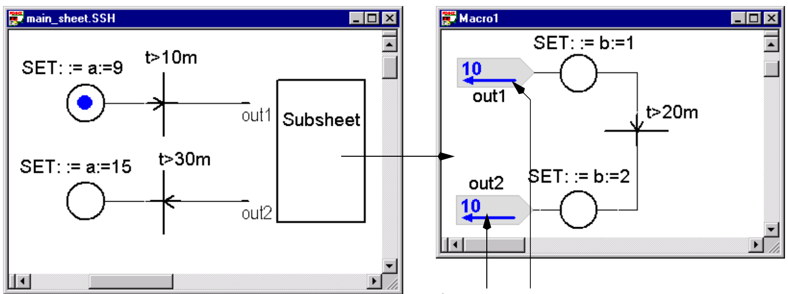
Marked components in the main sheet *Components in the subsheet* *Subsheet in the main sheet*

As in the variant above, the subsheet is considered a normal system component. To edit the subsheet, select the symbol, and then choose **«Open»** on the shortcut menu or choose SUBSHEET>OPEN.

State Graphs in Graphical Subsheets

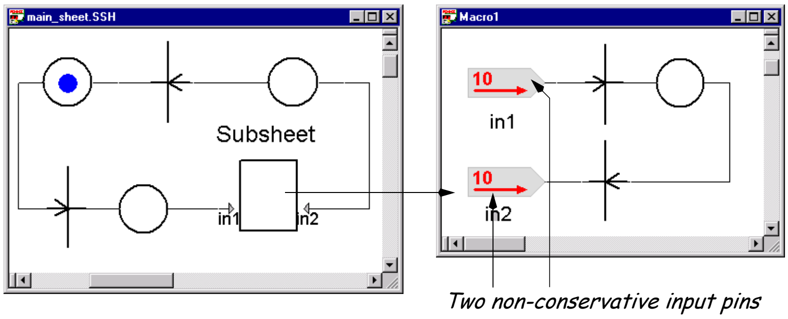
The input and output definitions in state graph structures in subsheet differs from the circuit (conservative nodes) and block diagram (input and output parameters).

If you want to create a state graph where **transitions** represent the subsheet edge, you must use **two non-conservative output pins** in the subsheet. Create two output pins and connect them with the state graph structure which starts with a state component.



Two non-conservative output pins

If you want to create a state graph where **states** represent the subsheet edge, you must use **two non-conservative input pins** in the subsheet. Create two input pins and connect them with the state graph structure which starts with a transition component.



Transferring Parameter Values into Graphical Subsheets

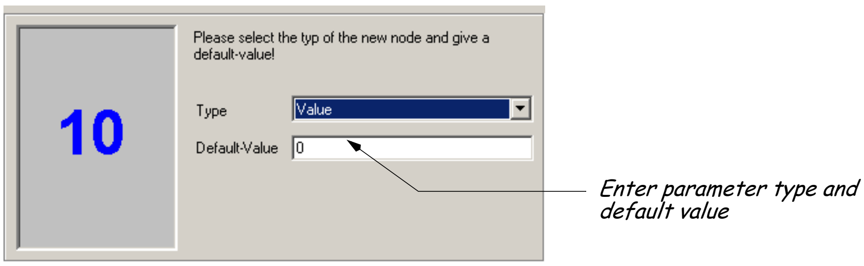
To use mainsheet parameters in the subsheet you need non-conservative input pins. These input pins are able to transfer a mainsheet quantity into the subsheet.

1 Choose SHEET>SUBSHEET OPEN to change into the subsheet wiring and choose **«Create Pin/Parameters...»** on the subsheet shortcut menu.

2 Select Non-Conservative Node and define a name. Click <Next> to continue.

3 Select **«Input»** and click <Next> to continue.

4 Select the pin type. Please note: The type does not influence the parameter itself. It select only a proper symbol which is displayed in the parameter dialogs.



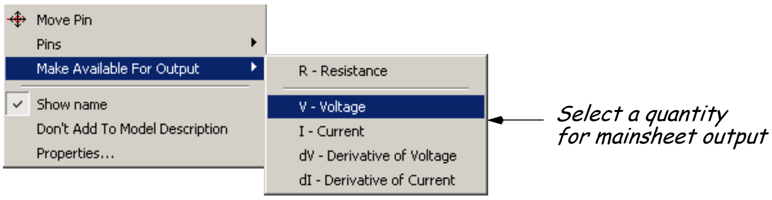
5 Select the pin position. If you do not click any pin, you have to create the parameter transfer with a name reference in the mainsheet.

6 The new pin symbol can now be connected with a parameter pin of a quantity in the subsheet. When no connection is used you can enter the pin name to transfer the mainsheet quantity.

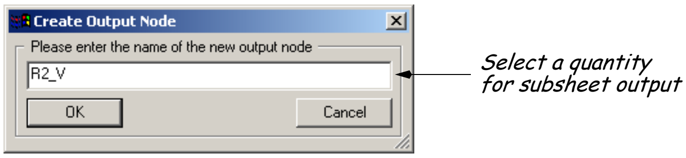
7 Choose **«Level Up»** in the subsheet shortcut menu to change to the next higher model level. Connect a mainsheet quantity with the pin or enter a name if the pin is disabled.

Transferring Parameters of Graphical Subsheets to the Mainsheet

1 Change into the subsheet wiring (SHEET>SUBSHEET OPEN) and make a subsheet quantity available for output. Click with the right mouse button a component or wire, select **«Make Available for Output»**, and choose one of the component quantities.



2 Enter the name for the output quantity. The name is used in the mainsheet.



3 Check the defined output quantities in the subsheet shortcut menu **«Edit Pin/Parameters…»**. You can see a list with all quantities available for the output in the mainsheet.

4 Choose **«Level Up»** on the subsheet shortcut menu to switch to the mainsheet. Double-click the subsheet symbol to open the subsheet property dialog. In the **«Output/Display»** dialog all output quantities are listed. Select a output format to display a quantity.

You can also use the SIMULATION>OUTPUTS command in the **mainsheet** to display subsheet quantities.

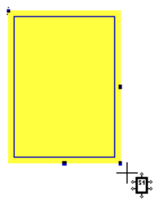
3.5.2 Text Subsheets

Text subsheets can be created in SML or VHDL-AMS description. To create a new SML subsheet choose SHEET>SUBSHEET>NEW SML20. To create a new VHDL-AMS subsheet choose SHEET>SUBSHEET>NEW VHDL-AMS. If a text subsheet command is started, the Text Editor is opened as embedded application. All functions as search and replace, syntax coloring, and syntax check are available within the Schematic window.

Creating Text Subsheets in SML Description

1 Choose SHEET>SUBSHEET>NEW SML20. The mouse pointer becomes a cross wire.

2 Draw the subsheet symbol to the size desired by holding the left mouse button and then release the mouse button. A rectangle for the macro is displayed and the empty text editor is opened.



3 Enter the subsheet description in SML. See also SML SIMPLORER Modeling Language (online help).

4 Define conservative nodes. Conservative nodes are connectors from the substructure to the next higher model level. They are defined in the following form: `PORT Nature_type: Node_name`, for example: *PORT electrical: N001*

5 Define non-conservative nodes. Non-conservative nodes are used to change parameters. They need the additional information whether the pin should be used as input, output, or input and output. They are defined in the following form:
`PORT data_type direction: Variable_name=Default_Value`, for example:
PORT real in: LM=0.1, PORT real out: V1=VM.V,PORT real inout: VAL=0

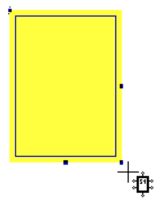
6 To check the SML syntax click the Scan Text symbol in the toolbar. If errors occur, an message box occurs on the screen.

7 Choose **«Level Up»** on the text editor shortcut menu (right.mouse click in the editor window) to switch to the mainsheet. Double-click the subsheet symbol to open the subsheet property dialog. In the **«Output/Display»** dialog all output quantities are listed. Select a output format to display a quantity.

Creating Text Subsheets in VHDL-AMS

1 Choose SHEET>SUBSHEET>NEW VHDL-AMS. The mouse pointer becomes a cross wire.

2 Draw the subsheet symbol to the size desired by holding the left mouse button and then release the mouse button. A rectangle for the macro is displayed and the empty text editor is opened.



3 Enter the subsheet description in VHDL-AMS. You can define only one model description in a subsheet.

4 Define model nodes. They are defined in the following form:
`PORT (QUANTITY Name: Data_type:=Default_value; TERMINAL Name : Nature_type)`, for example:
PORT (QUANTITY R : REAL: =1.0+3; TERMINAL p,m : ELECTRICAL)

5 To check the SML syntax click the Scan Text symbol in the toolbar. If errors occur, an message box occurs on the screen.

6 Choose **«Level Up»** on the Text Editor's shortcut menu (right mouse click in the editor window) to switch to the mainsheet. Double-click the subsheet symbol to open the subsheet property dialog. In the **«Output/Display»** dialog all output quantities are listed. Select a output format to display a quantity.

3.5.3 Subsheets in the Model Agent

In a simulation model defined subsheets (text or graphical) can be added to model libraries and so used in other sheets. You can insert subsheets from a model description or paste the copied subsheet directly in a model library.

Copying and Duplicating Subsheets

You can copy subsheets (CTRL+C) and insert them in the same or another sheet (CTRL+V). You can edit the subsheets after they are inserted.

In contrast, you can duplicate subsheets (EDIT>DUPLICATE or CTRL+D). All subsheets created with duplicate will always have the same content, because any changes are updated in all corresponding subsheets.

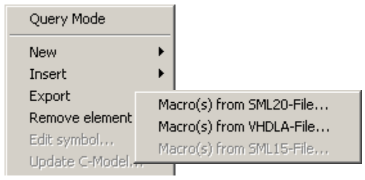
Creating Macros from Subsheet Models

Macros from Descriptions

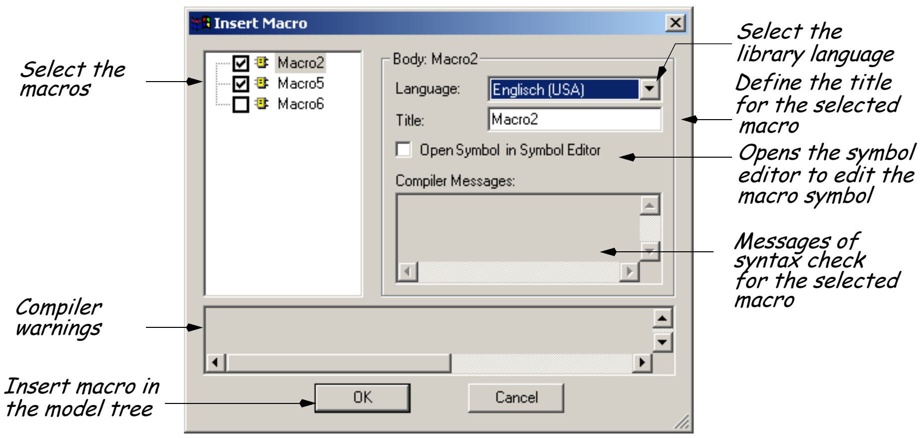
1 Create a simulation model with a subsheet element and run at least one simulation, or choose SIMULATION>EXPORT DESCRIPTION to create an .sml or .vhd file.

2 Select a library to store the new macro in the Model Agent.

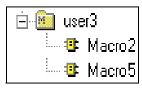
3 Click with the right mouse button in the pane above the libraries and choose **«Insert» «Macro(s) from SML-20 file»** or **«Macro(s) from VHDL-AMS file»** on the shortcut menu. Open the file (.sml or .vhd) of your simulation model in the dialog box.



The complete file is searched for macro definitions. Any subsheet that is found are suggested as a new macro model; the name can be specified.



4 Click <OK> to insert the subsheets as macros in the selected library.



5 Choose **«Edit Symbol»** on the macro shortcut menu to start the symbol editor to modify the default symbol. See also 16 “Symbol Editor” on page 499.

Macros from Subsheets

1 Choose EDIT>COPY or CTRL+C to copy a selected subsheet symbol in the sheet.

2 Go to the component tree, click a tab, choose a library, and click the folder where the new subsheet should be inserted.

3 Click with the right mouse the folder and choose **«Paste»** on the shortcut menu.

4 Choose **«Subsheet»** or **«Text Macro»**.

- Subsheet: You can drop the macro onto the sheet and open and edit the graphical presentation. This macro has and cyan colored symbol in the model tree.
- Text Macro: You can drop the macro onto the sheet, but you cannot open and modify the internal macro model. This macro has and yellow colored symbol in the model tree.

Please note: Text subsheets will always be added as a text macro to the model library.

3.6 Modifying Component Parameters

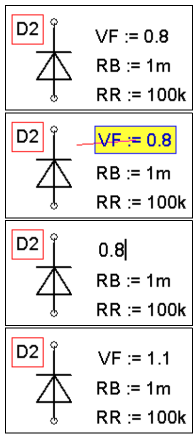
The names and parameters of components and simulation parameters in a simulation model can be modified with different tools and functions. You have the following choices:

- the Object Browser
- the property dialogs and input wizards of individual components
- the name and parameter displays on the sheet
- the EDIT>SEARCH/REPLACE command
- the EDIT>COPY DATA FROM... command

3.6.1 Editing Names and Parameters of Components on the Sheet

In addition to the Object Browser, the parameter dialogs and the component wizards, parameters can be edited directly on the sheet through the display elements. To do this, the name or parameter value must be displayed (click the element with the right mouse button, choose «Properties», and select the display settings). To edit in a display element, click the field twice (but not double-click). The edit mode is active, and you can modify the entries.

Editing Displays



1 For an element with displays for names and parameters, open the Properties or Parameters dialog to display names and parameters.

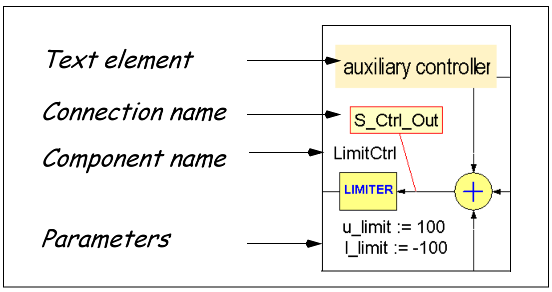
2 Click the display of a parameter. The name is highlighted in yellow.

3 Click a second time the display; the edit mode is active, and you can enter new value.

4 Click outside of the display. The modification is applied.

Starting Edit Mode with the F2 Key

The F2 key starts the edit mode for selected text elements on the sheet. Changes can then be entered directly on the sheet.



Fixed and User-Defined Model Parameters

Each SIMPLORER component is defined through its parameters. If you copy components's data, you must distinguish between fixed and user-defined model parameters.

- Fixed model parameters: values and initial values, among others, which belong to the component directly and cannot be deleted.
- User-defined model parameters: actions in states, formulas, and so on, which the user can define and delete.

Editing Parameters in the Property Dialog

Double-click a component on the model sheet, or select a component and choose EDIT>PROPERTIES to open the parameter dialog, where you can modify all values. See also “Defining Parameters” on page 49.

Editing Parameters in the Object Browser

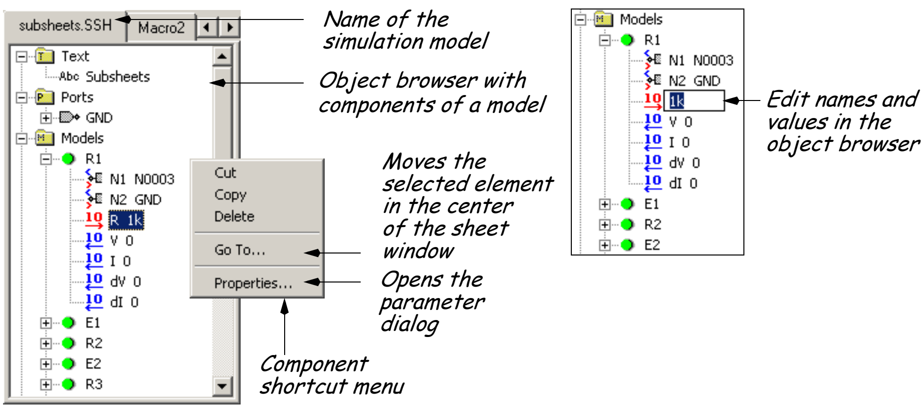
The Object Browser is displayed in a separate window and contains all components of a sheet and their properties. The structure is similar to the tree mode in Windows. The sequence of the elements can be changed by selecting an element and dragging it to a new location in the browser tree. Display elements can be divided into the following groups:

- Text
- Ports
- Models (name, parameters, pins, outputs)
- System Settings
- Nodes

Click a component name in the tree to view all of its parameters. To edit a parameter, select a separate entry of a component with another click. An edit field becomes active, allowing you to change the parameter.

Edit the component properties:

Click the component name with the right mouse button to open the shortcut menu. Choose «Properties» to open the parameter dialog. Click the name twice to start the edit mode (but do **not** double-click). You can change names and values directly in the Object Browser.

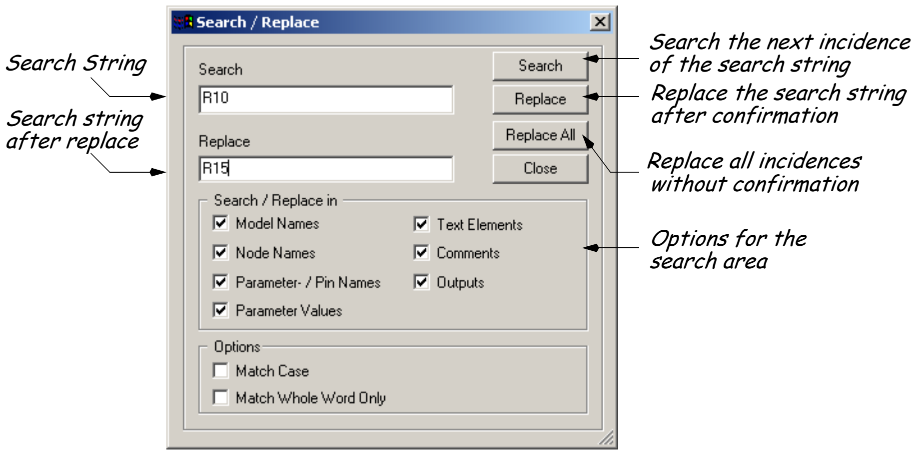


Search components on the sheet:
If you choose «Go to» on the shortcut menu of a component, it is moved to the center of the window. (You may want to zoom in to enlarge the wiring enough to position your selection more clearly.)

You can edit all model parameters, including simulation parameters and output quantities, in the Object Browser. You need not open the component property dialog to change parameters.

3.6.2 Searching and Replacing Parameters and Text

With the search and replace function (EDIT SEARCH/REPLACE) you can search for parameters and text in the model sheet and open the corresponding property dialog. Then you can define new names and values and change former settings as you wish.



The EDIT>SEARCH/REPLACE function searches the entire text; that is, each character combination is found regardless of its position in a word. The search can be restricted to a certain range with different options.

Text elements:	Search text elements which were created with the text tool.
Model names:	Search names of models, subsheets, and simulation parameters (only user-defined parameters can be changed).
Node names:	Search for node names.
Parameter/Pin names:	Search for parameter and pin names in models.
Parameter values:	Search for parameter values in models.
Comments:	Search for comments which are entered in the model's information field.
Outputs:	Search for defined outputs (cannot be changed).

Searching Text in Models and Dialogs

1 Choose EDIT SEARCH/REPLACE or press CTRL+F to start the search function.
2 Enter the search string and define the search field.
3 Start the search with <Search>.

For a successful search:

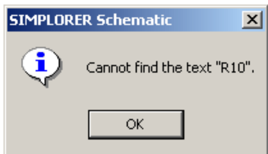
- The Information window shows the search results in the **«Build»** dialog. (Schematic automatically opens the Information window, if it is invisible.)
- The element with the searched character combination is selected.
- If the object browser is open, the element with the search string is selected in the object tree.

Before you open the property dialog, stop the search function with <Close>.

For an unsuccessful search:

You see this error message after an unsuccessful search. The search string is not available in the defined search field.

4 If necessary, continue the search with <Search> or stop searching with <Close>.



Error message after unsuccessful search

Replacing the Search String with New Characters

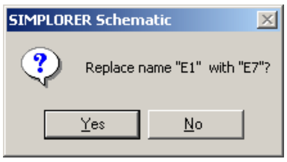
1 Start the search and replace function with EDIT>SEARCH/REPLACE.
2 Enter the search string and the new character combination in the corresponding fields.
3 Start the search and replace function with

<Search>

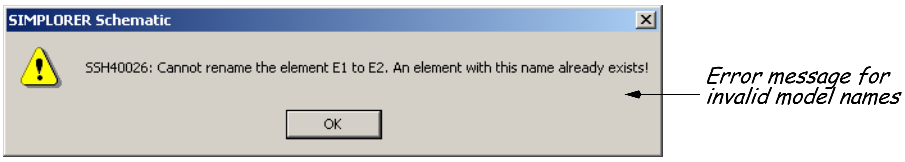
- The Information window shows the search results in the **«Build»** dialog. (Schematic automatically opens the Information window, if it is invisible.)
- The element with the searched character combination is selected.
- If the object browser is open, the element with the search string is selected in the object tree.

<Replace>

If you confirm the replacement at the prompt, the parameter and its links are updated.



If you try to apply an invalid name, an error message is displayed.



<Replace all>

All matching character strings found are replaced without confirmation. If the function tries to apply an invalid name for a model, an error message is displayed in the **«Build»** dialog of the Info window for all unreplaced names.

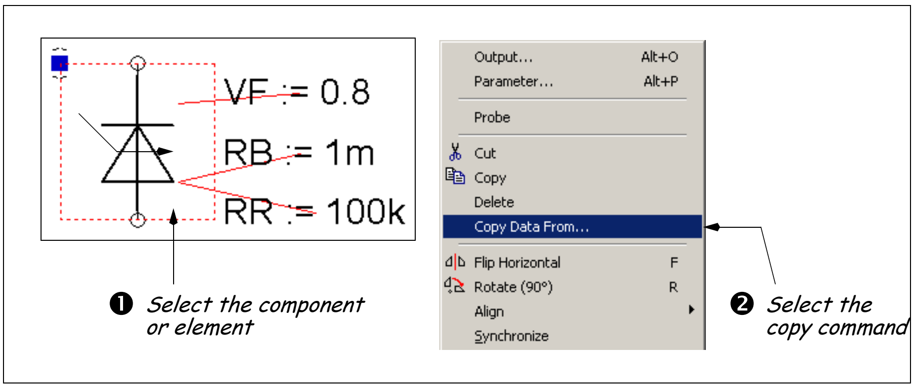
3.6.3 Transferring Parameters

The function **«Copy Data from»** quickly and accurately transfers parameters from one element to another element in the sheet. **«Copy Data from»** is available in the shortcut menu of an element or in the menu EDIT>COPY DATA FROM.

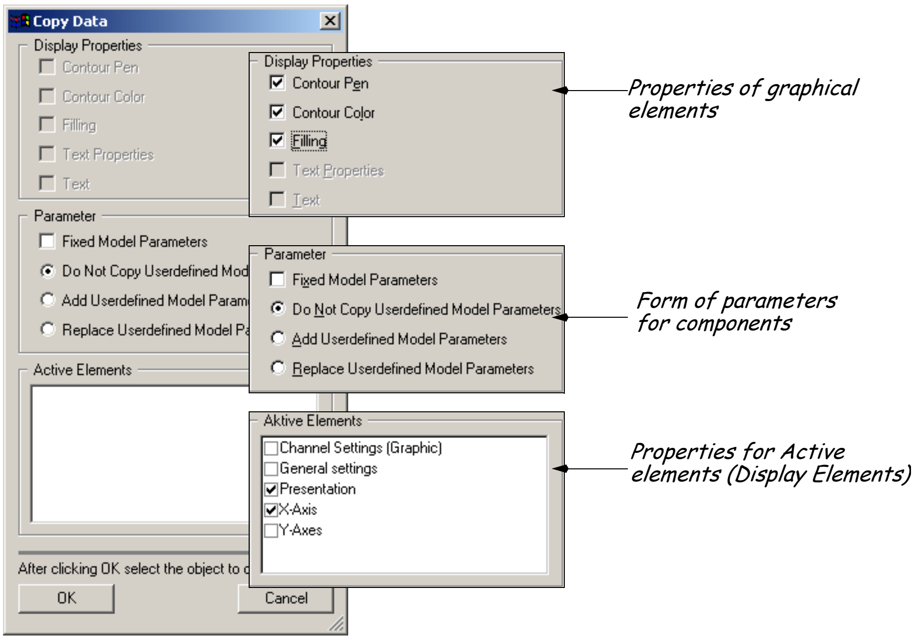
Transferring Parameters to another Components and Elements

1 Select the element which should receive the parameters.

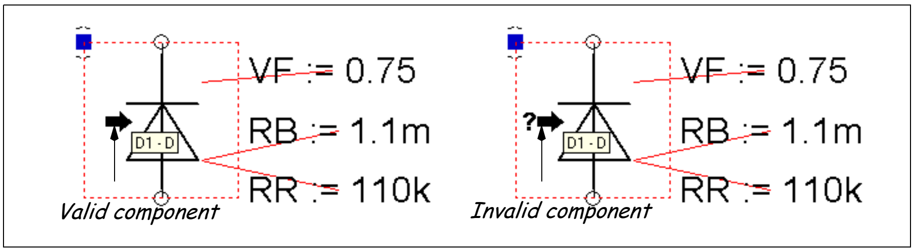
2 Choose **«Copy Data from»** on the shortcut menu or choose EDIT>COPY DATA FROM....



3 Choose the data type to be copied.



4 The mouse pointer converts into a large arrow with a question mark. Click the element from which data should be copied. The question mark disappears over a valid element.



You can also apply the function COPY DATA FROM to a multiple selection. All valid parameters are transferred from the initial element.

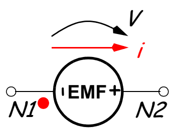
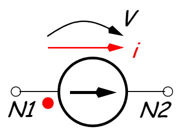
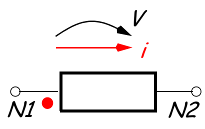
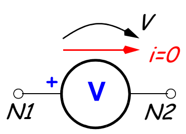
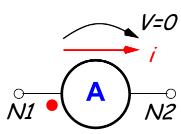
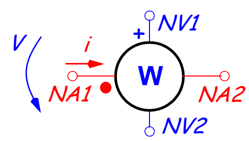
4 Modeling with Circuit Components

Electrical domain components are defined by an internal identifier, the name, their conservative nodes and a set of parameters. Terminals of conservative nodes cannot be deleted on the sheet. Electrical domain components appear in the *Circuit* folder of the «**Basics**» tab. Select the folder *Circuit*, choose one of the groups, click a name, drag the component onto the sheet and release the mouse button.

All forms of linear and nonlinear components can be used to model electrical circuits. However, only electrically reasonable circuits lead to correct simulation results. Voltage sources, current sources and switches are ideal components. Each independent circuit must be connected to a ground node at least once. See also “Network Configurations” on page 71.

Load Reference Arrow System of Circuit Components

The counting direction of current and voltage is marked by the red point or the plus sign at the symbol of electrical components.

	Voltage Sources	**Current Sources**	**Passive Components**
Electrical			
	Voltmeter	**Ammeter**	**Wattmeter**
Meters			

Network Configurations

In SIMPLORER only ammeters are allowed as controlling components for current controlled elements. These must be inserted properly in the controlling branch. If sources are part of mutual controlling sources in the circuit, stability problems may occur if the total gain of the loop is greater than or equal to one.

<u>The following types of network configurations are invalid:</u>

- Series connection of ideal current sources
- Series connection of inductors and ideal current sources
- Series connection of inductors with different initial values of current, $I0_1 \neq I0_2$
- Series connection of an inductor with an initial current value and an opened ideal switch or nonconducting system level semiconductor
- Parallel connection of ideal voltage sources
- Parallel connection of capacitors with ideal voltage sources
- Parallel connection of capacitors with different initial values of voltage, $V0_1 \neq V0_2$
- Meches which consist only of ideal sources (short-circuit)
- Open-ended branches

4.1 Passive Elements

- Resistor (R/RNL)
- Conductor (G/GVNL)
- Capacitor (C/CNL/CVNL)
- Inductor (L/LNL/LINL)
- Mutual Inductance (M)

Resistor

>>Basics>Circuit>Passive Elements

The component represents an electrical resistance. Within the dialog you can choose a linear or nonlinear resistor type.

Dialog Settings	
⊙	«Resistance [Ohm]» The Linear Resistor is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the resistance is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⊙	«Nonlinear i=f(v)» The Nonlinear Resistor is selected. Open the characteristic dialog or use the pin to define a characteristic. See also “Separate Component Characteristic” on page 297.
☑ ☐	«Use pin» If the box is checked, theresistor is defined by the characteristic connected to the pin, otherwise by the settings in the dialog <Characteristic>.

Linear Resistor

$$v(t) = R \cdot i(t)$$

The resistance is equivalent to the present value of the quantity specified in the text box. For the quantity any numerical value, variable, mathematical or logical expression can be used. Usually a Constant value, a Time function, or a variable are chosen, or the value of the resistance is determined in a state graph. A mathematical expression must conform to the syntax of the SIMPLORER expression interpreter. All system quantities and variables can be combined with mathematical and logical operators and constants.

However, the expression must be technically meaningful. If it is not, numerical instability or completely wrong results are possible. In particular, sudden changes in the first derivation of the control quantity should be avoided.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Resistance [Ω]	R	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Nonlinear Resistor

$i(t) = f(v(t))$

To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also “Separate Component Characteristic” on page 297.

X-value: voltage v (independent quantity)
Y-value: current i (dependent quantity)

To define characteristic data you can do one of the following:

- **«Use Pin»**; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:

- **«Reference»**; the name in the text box specifies an XY element placed on the sheet.
- **«Lookup Table»**

a. **«File Reference»**; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. **«Lookup Table»**; data pairs, defined in the table, are used for characteristic.

- ☑ **«File Name»;** if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic [/]	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Conductor

>>Basics>Circuit>Passive Elements

The component represents an electrical conductance. Within the dialog you can choose a linear or voltage-controlled conductor type.

Dialog Settings	
⊙	«Conductivity [S]» The Linear Conductor is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the conductivity is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⊙	«Nonlinear G=f(v)» The Nonlinear voltage-controlled Conductor is selected. Open the characteristic dialog or use the pin to define a characteristic. In addition, you have to specify a control component (quantity). See also “Separate Component Characteristic” on page 297.
☑ ☐	«Use pin» If the box is checked, the conductor is defined by the characteristic connected to the pin, otherwise by the settings in the dialog <Characteristic>.
	«Control Component» Define the control component (quantity). Quantity parameter type. The type accepts only a voltage of a circuit component (e.g. `E1.V`, `Name.V`). The values cannot be assigned by a variable. Enter the voltage quantity of a circuit component or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the control component is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Linear Conductor

$$i(t) = G \cdot v(t)$$

The conductivity is equivalent to the present value of the quantity specified in the **text box**. For the quantity any numerical value, variable, mathematical or logical expression can be used. Usually a Constant value, a Time function or a variable are chosen, or the value of the resistance is determined in a state graph. A mathematical expression must conform to the syntax of the SIMPLORER expression interpreter. All system quantities and variables can be combined with mathematical and logical operators and constants.

However, the expression must be technically meaningful. If it is not, numerical instability or completely wrong results are possible. In particular, sudden changes in the first derivation of the control quantity should be avoided.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Conductivity [S]	G	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Nonlinear voltage-controlled Conductor

$$i(t) = G(v(t)) \cdot v(t)$$

The voltage across the controlling component controls the conductivity corresponding to the specified characteristic function. To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also "Separate Component Characteristic" on page 297.

X-value: control voltage v (independent quantity)
Y-value: conductivity G (dependent quantity)

To define characteristic data you can do one of the following:

- **«Use Pin»**; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:

- **«Reference»**; the name in the text box specifies an XY element placed on the sheet.
- **«Lookup Table»**

a. **«File Reference»**; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. **«Lookup Table»**; data pairs, defined in the table, are used for characteristic.

- ☑ **«File Name»;** if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Control Quantity [V]	QUANT	real
Characteristic [/]	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Conductivity [S]	G	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real
Derivative of Charge [C/s]	dQ	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Capacitor

>>Basics>Circuit>Passive Elements

The component represents an electrical capacitance. Within the dialog you can choose a linear, nonlinear, or nonlinear dual capacitor type.

Dialog Settings	
⦿	«Capacitance [F]» The Linear Capacitor is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. You can also specify an initial voltage.
☑ ☐	«Use pin» If the box is checked, the capacitance is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⦿	«Nonlinear v=f(q)» The Nonlinear Capacitor is selected. Open the characteristic dialog or use the pin to define a characteristic. You can also specify an initial voltage. See also “Separate Component Characteristic” on page 297.
☑ ☐	«Use pin» If the box is checked, the capacitor is defined by the characteristic connected to the pin, otherwise by the settings in the dialog <Characteristic>.
⦿	«Nonlinear C=f(v)» The Nonlinear dual Capacitor is selected. Open the characteristic dialog or use the pin to define a characteristic. You can also specify an initial voltage. See also “Separate Component Characteristic” on page 297.
☑ ☐	«Use pin» If the box is checked, the capacitor is defined by the characteristic connected to the pin, otherwise by the settings in the dialog <Characteristic>.

	«Initial Value [V]» The initial voltage is valid for all capacitor types. Enter a numerical value, a variable, or an expression in the text box to define the initial voltage or use the pin to connect a quantity. The value is set only once at simulation start.
☑ ☐	**«Use pin»** If the box is checked, the initial voltage is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Linear Capacitor

$$i(t) = C \cdot \frac{dv(t)}{dt}$$

The capacitance is equivalent to the present value of the quantity specified in the text box. For the quantity any numerical value, variable, mathematical or logical expression can be used. Usually a Constant value, a Time function or a variable are chosen, or the value of the resistance is determined in a state graph. A mathematical expression must conform to the syntax of the SIMPLORER expression interpreter. All system quantities and variables can be combined with mathematical and logical operators and constants.

However, the expression must be technically meaningful. If it is not, numerical instability or completely wrong results are possible. In particular, sudden changes in the first derivation of the control quantity should be avoided.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Capacitance [F]	C	real
Initial Voltage [V]	V0	rea

Component Outputsl

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Charge [C]	Q	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real
Derivative of Charge [C/s]	dQ	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Nonlinear Capacitor

$$i(t) = \frac{dq(t)}{dt} \quad \text{with} \quad q(t) = f(v(t))$$

To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also “Separate Component Characteristic” on page 297.

X-value: charge q (independent quantity)
Y-value: voltage v (dependent quantity)

To define characteristic data you can do one of the following:

- «Use Pin»; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:

- «Reference»; the name in the text box specifies an XY element placed on the sheet.
- «Lookup Table»

a. «File Reference»; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. «Lookup Table»; data pairs, defined in the table, are used for characteristic.

- ☑ «File Name»; if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Initial Voltage [V]	V0	real
Characteristic	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Charge [C]	Q	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real
Derivative of Charge [C/s]	dQ	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Nonlinear dual Capacitor

$$i(t) = C(v(t)) \cdot \frac{dv(t)}{dt}$$

To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also “Separate Component Characteristic” on page 297.

X-value: voltage v (independent quantity)
Y-value: Capacitance C (dependent quantity)

To define characteristic data you can do one of the following:
- «Use Pin»; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:
- «Reference»; the name in the text box specifies an XY element placed on the sheet.
- «Lookup Table»

a. «File Reference»; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. «Lookup Table»; data pairs, defined in the table, are used for characteristic.

- ☑ «File Name»; if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

The primary form CNL of representing nonlinear capacitors leads in general to a better conditioned set of equations and should be preferred. With regard to an engineering model formulation, however, the dual form of representation may be more useful. Nevertheless, the application of the respective element type is up to the user. If numerical problems appear, an improvement may be obtained with the opposite implementation.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Initial Voltage [V]	V0	real
Characteristic [/]	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Inductor

>>Basics>Circuit>Passive Elements

The component represents an electrical inductance. Within the dialog you can choose a linear, nonlinear, or nonlinear dual inductor type.

4

Dialog Settings	
⦿	«Inductance [H]» The Linear Inductor is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. You can also specify an initial current.
☑ ☐	«Use pin» If the box is checked, the inductance is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⦿	«Nonlinear i=f(psi)» The Nonlinear Inductor is selected. Open the characteristic dialog or use the pin to define a characteristic. You can also specify an initial current. See also “Separate Component Characteristic” on page 297.
☑ ☐	«Use pin» If the box is checked, the inductor is defined by the characteristic connected to the pin, otherwise the settings in the dialog <Characteristic>.
⦿	«Nonlinear L=f(i)» The Nonlinear dual Inductor is selected. Open the characteristic dialog or use the pin to define a characteristic. You can also specify an initial current. See also “Separate Component Characteristic” on page 297.
☑ ☐	«Use pin» If the box is checked, the inductor is defined by the characteristic connected to the pin, otherwise by the settings in the dialog <Characteristic>.
	«Initial Value [A]» The initial current is valid for all types of inductor. Enter a numerical value, a variable, or an expression in the text box to define the initial current or use the pin to connect a quantity. The value is set only once at simulation start.
☑ ☐	«Use pin» If the box is checked, the initial current is defined by the quantity connected to the pin, otherwise the value in the text box is used.
☑	«Output Pin» If the box is checked, the component provides an output pin of the inductance value for connecting with a Mutual Inductance. You cannot use the normal parameter pin in this case because the mutual inductance requires an pin with the direction attribute input.

Linear Inductor

$$v(t) = \frac{d\Psi}{dt}$$

The inductance is equivalent to the present value of the quantity specified in the text box. For the quantity any numerical value, variable, mathematical or logical expression can be used. Usually a Constant value, a Time function or a variable are chosen, or the value of the resistance is determined in a state graph. A mathematical expression must conform to the syntax of the SIMPLORER expression interpreter. All system quantities and variables can be combined with mathematical and logical operators and constants.

However, the expression must be technically meaningful. If it is not, numerical instability or completely wrong results are possible.In particular, sudden changes in the first derivation of the control quantity should be avoided.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Inductance [H]	L	real
Initial Current [A]	I0	real
Inductance with Output Attribute [H]	LOUT	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Flux-linkage [PSI]	PSI	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real
Derivative of Flux-linkage [V]	dPSI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Nonlinear Inductor

$$v(t) = \frac{d\Psi}{dt} \qquad \text{with} \qquad \Psi = f(i(t))$$

To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also “Separate Component Characteristic” on page 297.

X-value: coupled flux ψ (independent quantity)
Y-value: current i (dependent quantity)

To define characteristic data you can do one of the following:
- **«Use Pin»**; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:
- **«Reference»**; the name in the text box specifies an XY element placed on the sheet.
- **«Lookup Table»**

a. **«File Reference»**; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. **«Lookup Table»**; data pairs, defined in the table, are used for characteristic.

- ☑ **«File Name»;** if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Initial Current [A]	I0	real
Characteristic [/]	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Flux Linkage [Vs]	PSI	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real
Derivative of Flux Linkage [V]	dPSI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Nonlinear dual Inductor

$$v(t) = L(i(t)) \cdot \frac{di(t)}{dt}$$

To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also “Separate Component Characteristic” on page 297.

X-value: current i (independent quantity)
Y-value: inductance L (dependent quantity)

To define characteristic data you can do one of the following:

- **«Use Pin»**; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:

- **«Reference»**; the name in the text box specifies an XY element placed on the sheet.
- **«Lookup Table»**

a. **«File Reference»**; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. **«Lookup Table»**; data pairs, defined in the table, are used for characteristic.

- ☑ **«File Name»;** if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

The primary form LNL of representing nonlinear inductors (see above) leads in general to a better conditioned set of equations and should usually be preferred. With regard to an engineering model formulation, however, the dual form representation may be more useful. Nevertheless, the application of the respective element type is up to the user. If numerical problems appear, an improvement may be obtained with the opposite implementation.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Initial Current [A]	I0	real
Characteristic	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Mutual Inductance

>>Basics>Circuit>Passive Elements

The component represents the mutual coupling between two inductors, L1 and L2, quantitatively determined by the factor *k*.

The mutual inductance is equivalent to the present value of the quantity specified in the text box. For the quantity any numerical value, variable, mathematical or logical expression can be used. Usually a Constant value, a Time function or a variable are chosen, or the value of the inductance is determined in a state graph. A mathematical expression must conform to the syntax of the SIMPLORER expression interpreter. All system quantities and variables can be combined with mathematical and logical operators and constants.

However, if the value of this quantity is unstable, numerical problems can occur. In particular, sudden changes in the first derivation of the control quantity should be avoided.

$$M = K \cdot \sqrt{L_1 \cdot L_2} \qquad \text{with} \qquad K = K_1 \cdot K_2$$

Dialog Settings	
	«Inductance L1/L2 [H]» L1 and L2 hold the names of coupled inductances. Quantity parameter type. The type accepts only an inductance parameter (e.g. `L1.L`, `Name.L`). The values cannot be assigned by a variable. Enter the quantity of a inductance or use the pin to connect the quantity.
☑ ☐	**«Use pin»** If the box is checked, the inductances are defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Mutual Inductance» K/M is the factor which quantitatively determines the mutual inductances. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	**«Use pin»** If the box is checked, the coupling factor or M is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Inductance L1 [H]	L1	real
Inductance L2 [H]	L2	real
Coupling Factor [/]	K	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Mutual Inductance [H]	M	real

4.2 Sources

- Voltage Source (E/ESINE/ETRIANG/EPULSE/EST_FALL/ESZ_RISE/ETRAPEZ)
- Controlled Voltage Source (EV/EVNL/EI/EINL)
- Current Source (I/ISINE/ITRIANG/IPULSE/IST_FALL/IST_RISE/ITRAPEZ)
- Controlled Current Source (I/IV/IVNL/II/IINL/ICNL)
- Voltage Controlled Oscillator (EVCO/IVCO)
- Polynomial Source (EPOLY/IPOLY)
- Fourier Source (/)
- Multidimensional Table Source (NDSRC)
- Power Source (P)

Voltage Source

>>Basics>Circuit>Sources

The component represents an independent voltage source. Within the dialog you can choose a constant, or the time-controlled sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling type.

Dialog Settings	
⊙	«EMF Value [V]» The Linear Voltage Source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the EMF value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
☑ ☐	«Partial Derivation» If the box is checked, the improved partial derivation algorithm is effective. In few cases, when the source value is defined with a equation, instabilities are possible. Clear the box to avoid the problem.

⦿	«Time Controlled» The predefined time-controlled voltage source is selected. In the list, select a time function and define the corresponding parameters. These are Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also “Separate Component Characteristic” on page 297.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.
☑ ☐	«Spice Compatible» If the box is checked, the spice compatible reference arrow system is applied to the source.
AC Parameters	
⦿	«Phase & Magnitude» The Phase/Magnitude representation is selected. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also “AC Simulation Models” on page 401.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension of the phase.
⦿	«Real & Imaginary» The Real/Imaginary representation is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Linear Voltage Source

$$v(t) = -EMF$$

The value of the electromotive force (EMF) is equivalent to the present value of the quantity specified in the text box. Usually a constant value, a time function, or a variable are chosen, or the value of the electromotive force is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

However, the expression must be technically meaningful. If it is not, numerical instability or completely wrong results are possible. In particular, sudden changes in the first derivation of the control quantity should be avoided.

The SIMPLORER voltage sources have the opposite orientation compared to SPICE-like voltage sources. See also “Load Reference Arrow System of Circuit Components” on page 71.

The current through an ideal voltage source can be arbitrary. Real voltage sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Component Parameters

Description [Unit]	Parameter Name	Data Type
EMF Value [V]	EMF	real
Partial Derivation Flag (1: On; 0: Off)	PARTDERIV	real
EMF Magnitude [V]	AC_MAG	real
EMF Phase Shift [deg]	AC_PHASE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
EMF Value [V]	EMF	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Time-Controlled Linear Voltage Source

$$v(t) = -f[\text{time function}]$$

The time controlled voltage sources have the same properties as the linear voltage source above; the current can be arbitrary and the sources have the opposite orientation compared to SPICE-like voltage sources. The value of the electromotive force instead is equivalent to the present value of the selected time function at each calculation step.

The simulator internal names of time controlled voltage sources read as follows: Sine – ESINE; Pulse – EPULSE; Triangular – ETRIANG; Trapezoidal – ETRAPEZ; Saw-tooth Rising – ESZ_RISE; Saw-tooth Falling – EST_FALL. The names are only relevant to simulation scripts in SML. Access to source parameters is carried out only by the component name.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [V]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, peridical function, PERIO=0; non-periodical function	PERIO	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real
EMF Magnitude [V]	AC_MAG	real
EMF Phase Shift [deg]	AC_PHASE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
EMF Value [V]	EMF	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Controlled Voltage Source

>>Basics>Circuit>Sources

The component represents a dependent voltage source. Within the dialog you can choose a voltage-controlled, current-controlled, nonlinear voltage-controlled, or nonlinear current-controlled type.

Dialog Settings	
▼	«Type» In the list, select the voltage-controlled, current-controlled, nonlinear voltage-controlled, or nonlinear current-controlled source.
	«Control Component» The voltage/current of the defined circuit component controls the value of the source. Quantity parameter type. The type accepts only voltage and current of voltmeters, ammeters, or wattmeters (e.g. `VM1.V`, `AM2.I`). The values cannot be assigned by a variable. Enter the quantity of a meter or use the pin to connect the quantity.
☑ ☐	«Use pin» If the box is checked, the control component is defined by the quantity connected to the pin, otherwise the name in the text box is used.
	«Factor» Only for linear controlled sources. The value of the control component is multiplied by the factor. The value '1' means the original value is used. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the factor is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Characteristic» Only for nonlinear controlled sources: EMF=f(v) -> voltage-controlled, EMF=f(i) -> current-controlled. Open the characteristic dialog or use the pin to define a characteristic. See also “Separate Component Characteristic” on page 297.
☑ ☐	«Use pin» If the box is checked, the source is defined by the characteristic connected to the pin, otherwise by the settings in the dialog <Characteristic>.

☑ ☐	«Spice Compatible» If the box is checked, the spice compatible reference arrow system is applied to the source.

Linear voltage/current-controlled Voltage Source

$$v(t) = -K \cdot v[\text{control component}]$$

$$v(t) = -K \cdot i[\text{control component}]$$

The value of the EMF is calculated from the voltage/current of the controlling circuit component multiplied by an arbitrary control coefficient *K*. For the control, any circuit component can be specified.

The SIMPLORER voltage sources have the opposite orientation compared to SPICE-like voltage sources. See also “Load Reference Arrow System of Circuit Components” on page 71.

In SIMPLORER only voltage sources or ammeters are allowed as controlling components for current controlled elements. These must be inserted properly in the controlling branch.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Control Quantity [V/A]	QUANT	real
Factor [/]	FACT	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Nonlinear voltage/current-controlled Voltage Source

$$v(t) = -f(v[\text{control component}])$$

$$v(t) = -f(i[\text{control component}])$$

The value of the EMF is calculated from the voltage/current of the controlling component corresponding to an arbitrary nonlinear function. The control can result from any component of the electrical circuit.

To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also “Separate Component Characteristic” on page 297.

X-value: control voltage/current (independent quantity

Y-value: EMF (dependent quantity)

To define characteristic data you can do one of the following:

- «Use Pin»; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:

- «Reference»; the name in the text box specifies an XY element placed on the sheet.
- «Lookup Table»

a. «File Reference»; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. «Lookup Table»; data pairs, defined in the table, are used for characteristic.

- ☑ **«File Name»;** if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

The SIMPLORER voltage sources have the opposite orientation compared to SPICE-like voltage sources. See also "Load Reference Arrow System of Circuit Components" on page 71.

In SIMPLORER only voltage sources or ammeters are allowed as controlling components for current controlled elements. These must be inserted properly in the controlling branch.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Control Quantity [V/A]	QUANT	real
Characteristic [/]	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Current Source

>>Basics>Circuit>Sources

The component represents an independent current source. Within the dialog you can choose a constant, or the time-controlled sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling type.

Dialog Settings	
⊙	«Value [A]» The Linear Current Source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.

☑ ☐	«Use pin» If the box is checked, the Source current is defined by the quantity connected to the pin, otherwise the value in the text box is used.
☑ ☐	«Partial Derivation» If the box is checked, the improved partial derivation algorithm is effective. In few cases, when the source value is defined with a equation, instabilities are possible. Clear the box to avoid the problem.
⦿	«Time Controlled» The predefined time-controlled current source is selected. In the list, select a time function and define the corresponding parameters. See also “Using Time Functions” on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.
AC Parameters	
⦿	«Phase & Magnitude» The Phase/Magnitude representation is selected. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also “AC Simulation Models” on page 401.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension of the phase.
⦿	«Real & Imaginary» The Real/Imaginary representation is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Linear Current Source

$$i(t) = I$$

The value of the source current is equivalent to the present value of the quantity specified in the text box. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

However, the expression must be technically meaningful. If it is not, numerical instability or completely wrong results are possible.In particular, sudden changes in the first derivation of the control quantity should be avoided.

The voltage of an ideal current source is arbitrary. Real current sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Source Current [A]	IS	real
Partial Derivation Flag (1: On; 0: Off)	PARTDERIV	real
EMF Magnitude [A]	AC_MAG	real
EMF Phase Shift [deg]	AC_PHASE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Source Current [V]	IS	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Time-Controlled Linear Current Source

$i(t) = f[\text{time function}]$

The time controlled current sources have the same properties as the linear current source above. The voltage of an ideal current source is arbitrary. The value of the source current instead is equivalent to the present value of the selected time function at each calculation step.

The simulator internal names of time controlled current sources read as follows: Sine – ISINE; Pulse – IPULSE; Triangular – ITRIANG; Trapezoidal – ITRAPEZ; Saw-tooth Rising – ISZ_RISE; Saw-tooth Falling – IST_FALL. The names only relevant to simulation scripts in SML. Access to source parameters is carried out only by the component name.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [A]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset [A]	OFF	real
Periodical; PERIO=1, peridical function, PERIO=0; non-periodical function	PERIO	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real
EMF Magnitude [A]	AC_MAG	real
EMF Phase Shift [deg]	AC_PHASE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Source Current [V]	IS	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Controlled Current Source

>>Basics>Circuit>Sources

The component represents a dependent current source. You can choose a voltage-controlled, current-controlled, nonlinear voltage-controlled, or nonlinear current-controlled type.

Dialog Settings	
▼	«Type» In the list, select the voltage-controlled, current-controlled, nonlinear voltage-controlled, nonlinear current-controlled type, or the Collector source.
	«Control Component» The voltage/current of the defined circuit component controls the value of the source. Quantity parameter type. The type accepts only voltage and current of voltmeters, ammeters, or wattmeters (e.g. `VM1.V`, `AM2.I`). The values cannot be assigned by a variable. Enter the quantity of a meter or use the pin to connect the quantity.
☑ ☐	«Use pin» If the box is checked, the control component is defined by the quantity connected to the pin, otherwise the name in the text box is used.
	«Factor» Only for linear controlled sources. The value of the control component is multiplied by the factor. The value '1' means the original value is used. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the factor is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Characteristic» Only for nonlinear controlled sources. IS=f(v) -> voltage-controlled, IS=f(i) -> current-controlled. Open the characteristic dialog or use the pin to define a characteristic. See also “Separate Component Characteristic” on page 297.
☑ ☐	«Use pin» If the box is checked, the voltage source is defined by the characteristic connected to the pin, otherwise by the settings in the dialog <Characteristic>.

	«Early Voltage [V]»/«Collector-Emitter potential [V]» Parameters of the Collector current source. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.

Linear voltage/current controlled Current Source

$$i(t) = K \cdot v[\text{control component}]$$

$$i(t) = K \cdot i[\text{control component}]$$

The value of the source current is calculated from the voltage/current of the controlling circuit component multiplied by an arbitrary control coefficient *K*. For the control, any circuit component can be specified.

In SIMPLORER only voltage sources or ammeters are allowed as controlling components for current controlled elements. These must be inserted properly in the controlling branch.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Control Quantity [V or A]	QUANT	real
Factor [/]	FACT	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Nonlinear voltage/current-controlled Current Source

$$i(t) = f(v[\text{control component}])$$

$$i(t) = f(i[\text{control component}])$$

The value of the source current is calculated from the voltage/current of the controlling component corresponding to an arbitrary nonlinear function. The control can result from any component of the electrical circuit.

To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also “Separate Component Characteristic” on page 297.

X-value: control voltage/current (independent quantity

Y-value: source current (dependent quantity)

To define characteristic data you can do one of the following:

- **«Use Pin»**; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:

- **«Reference»**; the name in the text box specifies an XY element placed on the sheet.
- **«Lookup Table»**

a. **«File Reference»**; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. **«Lookup Table»**; data pairs, defined in the table, are used for characteristic.

- ☑ **«File Name»;** if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

In SIMPLORER only voltage sources or ammeters are allowed as controlling components for current controlled elements. These must be inserted properly in the controlling branch.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Control Quantity [V/A]	QUANT	real
Characteristic	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Collector Current Source

$$i(t) = v_{be}(t) \cdot \ln\frac{v_{ce}(t) - VEA}{VCE - VEA}$$

The nonlinear collector current allows the static modeling of transistors (Early effect). The collector current depends via a nonlinear relation on both its own branch voltage ($v_{ce}(t)$) as well as the voltage of the controlling branch ($v_{be}(t)$).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Early Voltage [V]	VEA	real
Collector-emitter reference potential [V]	VCE	real
Control Quantity [V]	QUANT	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Voltage Controlled Oscillator

>>Basics>Circuit>Sources

The component represents a voltage controlled Oscillator. Within the dialog you can choose a voltage or current source.

Dialog Settings	
▼	«Type» In the list, select the Voltage or Current Source.
☐	«Limit Input Voltage» Each value of the input voltage, which controls the frequency of the voltage/current source, is accepted.
☑	«Limit Input Voltage» If the box is checked, you can enter an upper and lower limit in the corresponding text box. Values outside of the range are ignored. «Upper Limit»/«Lower Limit» Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
	«Output Amplitude» The value defines the amplitude of the sine wave. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
	«Phase» Phase shift of the created sine wave. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension. The dimension is valid for all angle input quantities in the VCO model.
☑ ☐	«Spice Compatible» If the box is checked, the spice compatible reference arrow system is applied to the source.

AC Parameters	
⊙	«Phase & Magnitude» The Phase/Magnitude representation is selected. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also “AC Simulation Models” on page 401.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension of the phase.
⊙	«Real & Imaginary» The Real/Imaginary representation is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Voltage Controlled Oscillator Voltage and Current Source

$$
\begin{aligned}
v(t) &= \mathrm{Ampl} \cdot \sin(2\pi \cdot f + \mathrm{Phase}) \\
f &= f(v_{control}) \\
f &= \mathrm{Value}(UL) \quad \text{when} \quad UL < v_{control} \\
f &= \mathrm{Value}(LL) \quad \text{when} \quad LL > v_{control}
\end{aligned}
$$

The VCO provides a sine wave with a frequency dependent on the input signal. The frequency is changed in relation 1:1, this means: $v_{control}$=50V -> f=50Hz, … Furthermore you can define amplitude and phase shift of the output sine wave and limit of the control signal.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Voltage Upper Limit [V]	UL	real
Input Voltage Lower Limit [V]	LL	real
Amplitude	AMPL	real
Phase [deg]	PHASE	real
EMF Magnitude [V or A]	AC_MAG	real
EMF Phase Shift [deg]	AC_PHASE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Frequency	FREQU	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Polynomial Source

>>Basics>Circuit>Sources

The component represents a polynomial source. Within the dialog you can choose a voltage or current source.

The function models polynomials of free order, specified by the polynomial coefficients $p_0 \ldots p_n$. The meaning of the coefficients depends on the order of the polynomial.

onedimensional polynom m=1

$$x_a(t) = f(x_e(t)) = p_0 + p_1 x_e + p_2 x_e^2 + p_3 x_e^3 + \ldots + p_k x_e^k$$

twodimensional polynom m=2

$$x_a(t) = f(x_{e_1}(t), x_{e_2}(t)) = p_0 + p_1 x_{e_1} + p_2 x_{e_2} + p_3 x_{e_1}^2 + p_4 x_{e_1} x_{e_2} + p_5 x^2 e_2 + p_6 x_{e_1}^3$$
$$+ p_7 x_{e_1}^2 x_{e_2} + p_8 x_{e_2} x_{e_2}^2 + p_9 x_{e_2}^3 + p_{10} x_{e_1}^4 + p_6 x_{e_1}^3$$

threedimensional polynom m=3

$$x_a(t) = f(x_{e_1}(t), x_{e_2}(t), x_{e_3}(t)) = p_0 + p_1 x_{e_1} + p_2 x_{e_2} + p_3 x_{e_3} + p_4 x_{e_1}^2 + p_5 x_{e_1} x_{e_2} + p_6 x_{e_1} x_{e_3}$$
$$+ p_7 x_{e_2}^2 + p_8 x_{e_2} x_{e_3} + p_9 x_{e_3} + p_{10} x_{e_1}^3 + p_{11} x_{e_1}^2 x_{e_2} + p_{12} x_{e_1}^2 x_{e_3} + p_{13} x_{e_1} x_{e_2}^2 + p_{14} x_{e_1} x_{e_2} x_{e_3}$$
$$+ p_{15} x_{e_1} x_{e_3}^2 + p_{16} x_{e_2}^3 + p_{17} x_{e_2}^2 x_{e_3} + p_{18} x_{e_2} x_{e_3}^2 + p_{19} x_{e_3}^3 + p_{20} x_{e_1}^4$$

with $x_a(t)$ = output signal, $x_{e_1 \ldots e_n}(t)$ = input signal(s), $n=1 \ldots m$

Dialog Settings	
▼	«Source Type» In the list, select Voltage or Current Source.
▲▼	«Order» Defines the order of the polynom. Please note, you cannot define a variable or an expression in this case. Only a positive whole number is accepted.
	«Control Quantity» Define the control component (quantity). Quantity parameter type. The type accepts only voltage and current of voltmeters, ammeters, or wattmeters (e.g. `VM1.V`, `AM2.I`). The values cannot be assigned by a variable. Enter the quantity of a meter or use the pin to connect the quantity.
	«Coefficient» Define the Polynomial coefficients. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
☑ ☐	«Spice Compatible» If the box is checked, the spice compatible reference arrow system is applied to the source.

AC Parameters	
⦿	«Phase & Magnitude» The Phase/Magnitude representation is selected. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also “AC Simulation Models” on page 401.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension of the phase.
⦿	«Real & Imaginary» The Real/Imaginary representation is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Polynomial Voltage and Current Source

Voltage-controlled source:

$$i_a(t) = f(v_{el}(t) \ldots v_{en}(t)) \quad \text{with} \quad n=1 \ldots m \quad \text{m Dimension of the polynom}$$
$$v_a(t) = f(v_{el}(t) \ldots v_{en}(t)) \quad \text{with} \quad n=1 \ldots m$$

Current-controlled source:

$$i_a(t) = f(i_{el}(t) \ldots i_{en}(t)) \quad \text{with} \quad n=1 \ldots m \quad \text{m Dimension of the polynom}$$
$$v_a(t) = f(i_{el}(t) \ldots i_{en}(t)) \quad \text{with} \quad n=1 \ldots m$$

Component Parameters

Description [Unit]	Parameter Name	Data Type
Maximum of Power [/]	ORD	real
Control Components and Quantity [V or A]	QUANT[n]	real
Coefficients [/]	COEFF[n]	real
EMF Magnitude [V or A]	AC_MAG	real
EMF Phase Shift [deg]	AC_PHASE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
EMF Value [V] (voltage source)	EMF	real
Source Current [A] (current source)	IS	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Fourier Source

>>Basics>Circuit>Sources

The component represents a Fourier source. The EMF value of the Fourier voltage source results from the ripple amplitude of the defined sine waves. The values can be imported from a .mdx file created in the DAY Post Processor through an FFT analysis. In addition, you can also define the fundamental harmonics and the number of sub harmonics in the component dialog. Within the dialog you can choose a voltage or current source.

The component cannot be used with AC and DC simulation.

Dialog Settings	
▼	«Source Type» In the list, select Voltage or Current Source.
⊙	«Harmonics» Select All, Even or Odd harmonics. The setting is active in the table on the right hand side. When you the define the **«Number»** of harmonics all or only the even or odd harmonics are considered.
☑ ☐	«Phase» If the box is deactivated, you can define your own values in the corresponding text box on the right hand side. Click in the list field and enter the phase value either in degrees or in radians.
	«Ampl.»«Ampl. [%]» Click the corresponding field in the list and type the value of the amplitude either as absolute or proportional value. Please note, you cannot define a variable or an expression in this case. Only a number is accepted.
	«Fundamental Frequency» Defines the fundamental frequency. This value is the basis of all harmonics listed in the table.
	«Number» Defines the number of harmonics considered in the simulation. Please note, you cannot define a variable or an expression in this case. Only a number is accepted.
☑ ☐	«File Name» If the box is checked, you can define a .mdx file with the corresponding data. See also 14.2 "SIMPLORER ASCII System Data Formats" on page 467.

Fourier Voltage and Current Source

Voltage Source

$$v(t) = \sum_{\nu = n}^{m} A_\nu \cdot \sin(2\pi \cdot \nu \cdot f_1 \cdot t + \varphi_\nu)$$

Current Source

$$i(t) = \sum_{\nu = n}^{m} A_\nu \cdot \sin(2\pi \cdot \nu \cdot f_1 \cdot t + \varphi_\nu)$$

Component Parameters

Description [Unit]	Parameter Name	Data Type
Maximum of Power [/]	ORD	real
Control Components and Quantity [V or A]	QUANT[n]	real
Coefficients [/]	COEFF[n]	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
EMF Value [V] (voltage source)	EMF	real
Source Current [A] (current source)	IS	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Multidimensional Table Source

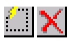
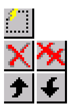
>>Basics>Circuit>Sources

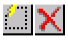
The multidimensional table source represents a table of controlled electrical sources (voltage and current) which depend on different, nonlinear input quantities. Each input quantity (node voltage or current of ammeter) has a specific characteristic with extrapolation type. The characteristics of input values can also be imported from a .mdx file created in the DAY Post Processor. Eeach defined source has two pins, whereas pin 0 and pin 1 belong to the first source, pin 2 and pin 3 to the seond source, an so on. Defined voltage sources have at the second pin the red point (N2 of standard voltage sources), current sources have at the first pin the red point (N1 of standard current sources).

$$v[n](t) = -f(v(QUANT)[m], i(QUANT)[m])$$

$$i[n](t) = f(v(QUANT)[m], i(QUANT)[m])$$

Dialog Settings	
	«Name» Displays the component internal names of the input quantities which are defined in the <Create/Edit Table> dialog. You cannot make changes in these fields.
	«Input» The voltage/current of the defined quantities control the values of the sources. Quantity parameter type. The type accepts only a node potential or the current of an ammeters (e.g. `N0003.V`, `AM2.I`). The values cannot be assigned by a variable. Enter the quantity or use the pin to connect the quantity.
☑ ☐	«Use pin» If the box is checked, the control signal is defined by the quantity connected to the pin, otherwise the name in the text box is used.

▼	«Extrapolation» In the list, select the **«Linear»**, **«Periodic»**, **«Halfperiod»**, **«Constant»**, **«Even»**, or **«Odd»** extrapolation type to define the quantity characteristics outside the range of parameter sweep. The default extrapolation is **«Linear»**. Select **«Linear»** to extrapolate a linear waveform. This option takes the last two points of the wave and generates a straight line extending beyond the range of parametric sweep. Select **«Periodic»** to repeat the wave outside the range of parametric sweep. Select **«Halfperiod»** to mirror then repeat the waveform outside the range of parameter sweep. Select **«Constant»** to extrapolate a constant value from the last point in the interpolation. Select **«Even»** to repeat the wave outside the range of parametric sweep. Select **«Odd»** to repeat a reflection of the waveform outside the range of the parametric sweep. For example, extrapolating with this option reflects the wave into the positive range.
▼	«Interpolation» In the list, select the «NOSPLINE», «LASTSPLINE», or «DEEPSPLINE» interpolation type to define the quantity characteristic inside the range of parameter sweep. The default interpolation is «NOSPLINE». The interpolation type is valid for all input quantities. Select «NOSPLINE» to fit a straight line between the points of interpolation. Select «LASTSPLINE» to apply a Bezier interpolation to the last quantity characteristic. Select «DEEPSPLINE» to apply a Bezier interpolation to all quantity characteristics.
☑ ☐	«File Name» If the box is checked, characteristics of input values are defined by the file entered in the text box, otherwise the values in the <Create/Edit Table> dialog are used. Please note: Select the correct file filter (.mdx) before browsing.
	«Name» Displays the component internal names of the output quantities which are defined in the <Create/Edit Table> dialog. You cannot make changes in these fields.
▼	«Type» In the list field, select Voltage (VSRC) or Current (CSRC) source for output.
	<Create/Edit Table> Click to open the dialog to create or edit input and output quantities.
Input Quantities	
	«Edit Input Channels» Create or delete entries with the symbols placed on the upper right side.
	«Index» Displays the component internal index of the input quantities. You cannot make changes in these fields.
	«Name»«Unit» Click in the fields to define names and units for the corresponding input quantity. Both are displayed in the column headings of the lookup table and saved in the created .mdx file.
	«Channel Data» Select an input quantity in the table to define the corresponding characteristic data in the lookup table. The name of the active input quantity displays the group box inscription. Create, delete or move the entries with the symbols placed on the upper right side. Click in the fields to enter values. Please note, you cannot use variables or expressions. Only numbers are accepted.

☑ ☐	«Sweep Data» If the box is checked, you can sweep data for a given range of values. <Set> replaces the values in the lookup table, <Append> inserts the new data at the table end.
Output Quantities	
	«Edit Output Channels» Create or delete entries with the symbols placed on the upper right side. Each new source has two conservative nodes (pins). The first source has pin 0 and 1, the second source 2 and 3, …
	«Index» Displays the component internal index of the output quantities. You cannot make changes in these fields.
	«Name»«Unit» Click in the fields to define names and units for the corresponding input quantity. Both are displayed in the column headings of the lookup table and saved in the created .mdx file.
	«Edit Output Data» Click in the fields to enter values for corresponding output sources. You can also use the <Tree> mode. Both views are automatically updated. Please note, you cannot use variables or expressions. Only numbers are accepted.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Table File Name	P	real
Interpolation Type	IP	real
Control Input	QUANT[n]	real
Extrapolation Type	EP[n]	real
Source Type	SRC[n]	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Output Vector	SRC_QUANT[n]	real
Derivative of Output Vector	dSRC_QUANT[n][m]	real

Component Nodes

Description	Node Name	Nature
First Source	N0 N1	electrical
Second Source	N2 N3	electrical

Power Source

>>Basics>Circuit>Sources

The component represents a power source able to work as source (positve power) or sink (negative power). However, the circuit must be technically meaningful. You cannot use the component as sink when it is connected only with a resistor. If you use the component as source (positive power) connected only with a resistor, two solutions with opposite current and voltage values exist.

$$POWER(t) = v(t) \cdot i(t)$$

Dialog Settings	
	«Power» Defines the provided or consumed power. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the control signal is defined by the quantity connected to the pin, otherwise the name in the text box is used.
AC Parameters	
⊙	«Phase & Magnitude» The Phase/Magnitude representation is selected. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also “AC Simulation Models” on page 401.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension of the phase.
⊙	«Real & Imaginary» The Real/Imaginary representation is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Power [W]	P	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

4.3 Switches

- Ideal Switch (S)
- Ideal Transfer Switch (TS)
- Controlled Switch (CSV/CSI)

Ideal Switch

>>Basics>Circuit>Ideal Switches

The component represents an ideal electrical switch. If the control signal is greater than '0', the two terminal nodes of the switch are connected. Because it is an ideal switch, the resistance of the line connection is zero. If the control signal is lower than or equal to '0', the line is disconnected and the resistance is infinite. A logical signal controls the state of the switch.

Since the switch behaves as an ideal, physically meaningful circuits must be obtained in both ON and OFF switching states.

The ideal switch is represented as an Animated Symbol; that means the switch symbol will change during the simulation depending on the control signal. To display the states with the switch, check the **«Animated Symbols»** box in the «System» tab in SHEET>PROPERTIES.

$$v(t) = 0 \quad \text{if the control signal } > 0 \text{ (On)}$$
$$i(t) = 0 \quad \text{if the control signal } \leq 0 \text{ (Off)}$$

Dialog Settings	
	«Control Signal» Defines the switch control signal. Control parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the control signal is defined by the quantity connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Control Signal	CTRL	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical

Ideal Transfer Switch

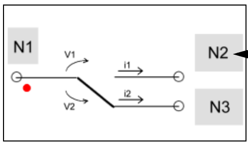
>>Basics>Circuit>Ideal Switches

The component represents an ideal electrical transfer switch. If the control signal is greater than '0', the nodes N1 and N2 are connected and the nodes N1 and N3 are disconnected. Because it is an ideal switch, the resistance of the line connection is zero and the resistance of the disconnection is infinite. If the control signal is lower than or equal to '0', the line between I and N3 is connected.

Since the switch behaves as an ideal, physically meaningful circuits must be obtained in both ON and OFF switching states.

The ideal switch is represented as an Animated Symbol; that means the switch symbol will change during the simulation depending on the corresponding control signal. To display the states with the switch, check the **«Animated Symbols»** box in the «System» tab in SHEET>PROPERTIES.

$$v_1(t) = 0 \quad \text{and} \quad i_2(t) = 0\text{, if the control signal } > 0 \text{ (On)}$$
$$i_1(t) = 0 \quad \text{and} \quad v_2(t) = 0\text{, if the control signal } \le 0 \text{ (Off)}$$



The start configuration of the transfer switch is Off, as represented in the figure

Dialog Settings	
	«Control Signal» Defines the switch control signal. Control parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the control signal is defined by the quantity connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Control Signal	CTRL	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical
Pin 3	N3	electrical

Controlled Switch

>>Basics>Circuit>Ideal Switches

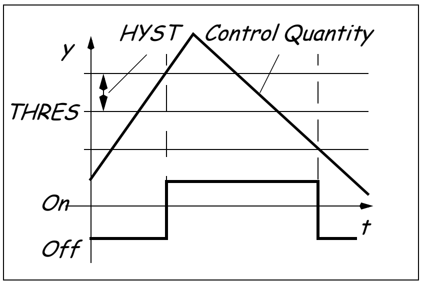
The component represent an ideal electrical switch. Within the dialog you can choose a voltage or current controlled switch.

If the control quantity (V or I) is greater than threshold plus hysteresis, the controlled switch is on. If the control quantity is lower than threshold minus hysteresis, the controlled switch is off. Because it is an non-ideal switch, you can define an On and Off resistance for the line connection.

The component cannot be used with AC and DC simulation.

You cannot use zero for the on resistance.

Off $\rightarrow$ On	V or I > (THRES + HYST)
On $\rightarrow$ Off	V or I < (THRES – HYST)



Dialog Settings	
▼	«Type» In the list, select Voltage or Current controlled switch.
	«On State Resistance»«Off State Resistance» Defines the resistance for the on and off state. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
	«Threshold Voltage»«Threshold Current» Defines the threshold of the controlled switch. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Hysteresis Voltage»«Hysteresis Current» Defines the hysteresis of the controlled switch. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Initial State» Defines the initial state of the controlled switch. Enter a numerical value, a variable, or an expression in the text box to define the initial state at simulation start.

Component Parameters

Description [Unit]	Parameter Name	Data Type
On State Resistance [Ω]	RON	real
Off State Resistance [Ω]	ROFF	real
Threshold Voltage/Current [V]/[I]	VTHRES/ITHRES	real
Hysteresis Voltage/Current [V]/[I]	VHYST/IHYST	real
Initial State, 0=OFF 1=ON [/]	CTRL0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Switch Pin 1 (with red point)	N1	electrical
Switch Pin 2	N2	electrical
Contol Voltage/Current Pin 1 (with blue cross/red point)	NC1	electrical
Contol Voltage/Current Pin 2	NC2	electrical

4.4 Semiconductors System Level

- Diode (D/DEQUL/DEXP/DXY)
- IGBT (IGBT/IGBTEQUL/IGBTEXP)
- MOS (MOS/MOSEQUL/MOSEXP)
- BJT (BJT/BJTEQUL/BJTEXP)
- GTO (GTO/GTOEQUL/GTOEXP)
- TH (TH/THEQUL/THEXP)
- TRIAC (TRIAC/TRIACEQUL/TRIACEXP)

Diode

>>Basics>Circuit>Semiconductors System Level

The System level Diode model is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the selected characteristic.

The diode types of predefined characteristics defined within the component dialog guarantee a high calculation speed. In contrast to the predefined characteristic types (equivalent line, exponential function, lookup table) you can use each other separate characteristic from the *Tools* folder or an appropriate quantity of a component.Characteristics based on Lookup tables allow simple access to measured curves for the switching behavior of the components.

```
v(t) = f(i(t))
```

Dialog Settings	
⊙	«Characteristic Type» The diode type of predefined characteristics is selected. In the list, select a characteristic.
▼	«Type» In the list, select the equivalent line, exponential function, or XY Data Pairs and define the corresponding characteristic parameters. These are Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
	Equivalent Line (DEQUL): The diode characteristic is defined by an equivalent line. You have to define a value for the Forward voltage, Bulk resistance and Reverse resistance. V_{limit} stands for the intersection of the forward and reverse characteristic and is calculated by the program itself. $I = \frac{v(t) - V_f}{R_b}$ for $v(t) \ge V_{limit}$ $I = \frac{v(t)}{R_r}$ for $v(t) < V_{limit}$ Exponential Function (DEXP): The diode characteristic is defined by an Exponential function. You have to define values for the Saturation current, Thermal voltage and Reverse resistance. $i(t) = I_s \cdot \left(e^{\frac{v(t)}{V_t}} - 1\right)$ for $v(t) \ge 0$ $i(t) = \frac{v(t)}{R_r}$ for $v(t) < 0$ XY Data Pairs (DXY) The diode characteristic is defined by a data table. You can define the table within the characteristic dialog directly or load an existing file. The data pairs are displayed in the lookup table and used for the characteristic. You can modify the data when necessary. See also “Separate Component Characteristic” on page 297.
⊙	«Element Name» The characteristic of the diode is defined by a characteristic component. Characteristic parameter type. Enter the name in the text box (e.g. `XY1.VAL`, `EXP.VAL`) or use the pin to connect the characteristic component. The internal used type is D.
☑ ☐	«Use pin» If the box is checked, the name is defined by the characteristic connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic; D	CH	real
Forward Voltage [V]; DEQUL	VF	real
Bulk Resistance [Ω]; DEQUL	RB	real
Reverse Resistance [Ω]; DEQUL/DEXP	RR	real
Saturation Current [A]; DEXP	ISAT	real
Thermal Voltage [V]; DEXP	VT	real
Characteristic; DXY	FILE	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Conductivity [S]	G	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Anode	A	electrical
Cathode	C	electrical

IGBT

>>Basics>Circuit>Semiconductors System Level

The System level IGBT (Insulated Gate Bipolar Transistor) model is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the selected characteristic.

The IGBT types of predefined characteristics defined within the component dialog guarantee a high calculation speed. In contrast to the predefined characteristic types (equivalent line, exponential function) you can use each other separate characteristic from the *Tools* folder or an appropriate quantity of a component. Usually an Equivalent line, Exponential function, Hyperbolic function, Polynom or the XY Lookup Table should be used. Characteristics based on XY-Data pairs allow simple access to measured curves for the switching behavior of the components.

To control the IGBT, a logical signal is applied. This signal can originate from any SIMPLORER system variable.

Control Signal (Base current)	Component State
> 0	ON
≤ 0	OFF

$$v(t) = f(i(t), [\text{control signal}])$$

Dialog Settings	
⊙	«Characteristic Type» The IGBT type of predefined characteristics is selected. In the list, select a characteristic.
▼	«Type» In the list, select the equivalent line, exponential function, or XY Data Pairs and define the corresponding characteristic parameters. These are Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Equivalent Line (IGBTEQUL):
The IGBT characteristic is defined by an equivalent line. You have to enter a value for the Forward voltage, Bulk resistance and Reverse resistance.

V_{limit} stands for the intersection of the forward and reverse characteristic and is calculated by the program itself.

$$I = \frac{v(t) - V_f}{R_b} \quad \text{for} \quad v(t) \geq V_{limit}$$

$$I = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < V_{limit}$$

Exponential Function (IGBTEXP):
The IGBT characteristic is defined by an Exponential function. You have to enter values for the Saturation current, Thermal voltage and Reverse resistance.

$$i(t) = I_s \cdot \left(e^{\frac{v(t)}{V_t}} - 1 \right) \quad \text{for} \quad v(t) \geq 0$$

$$i(t) = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < 0$$

XY Data Pairs (IGBTXY)
The IGBT characteristic is defined by a data table. You can define the table within the characteristic dialog directly or load an existing file. The data pairs are displayed in the lookup table and used for the characteristic. You can modify the data when necessary. See also “Separate Component Characteristic” on page 297.

⊙ **«Element Name»**
The characteristic of the IGBT is defined by a characteristic component. Characteristic parameter type. Enter the name in the text box (e.g. `XY1.VAL`, `EXP.VAL`) or use the pin to connect the characteristic component. The internal used type is **IGBT**.

☑ ☐ **«Use pin»**
If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.

«Control Signal»
Defines the IGBT control signal. Control parameter type. Enter Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.

☑ ☐ **«Use pin»**
If the box is checked, the name is defined by the characteristic connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic; IGBT	CH	real
Control Signal; All components	CTRL	real
Forward Voltage [V]; IGBTEQUL	VF	real
Bulk Resistance [Ω]; IGBTEQUL	RB	real
Reverse Resistance [Ω]; IGBTEQUL/IGBTEXP	RR	real
Saturation Current [A]; IGBTEXP	ISAT	real
Thermal Voltage [V]; IGBTEXP	VT	real
Characteristic; IGBTXY	FILE	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Conductivity [S]	G	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Collector	C	electrical
Emitter	E	electrical

MOS Fieldeffect Transistor

>>Basics>Circuit>Semiconductors System Level

The System level MOS model is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the selected characteristic.

The MOS types of predefined characteristics defined within the component dialog guarantee a high calculation speed. In contrast to the predefined characteristic types (equivalent line, exponential function) you can use each other separate characteristic from the *Tools* folder or an appropriate quantity of a component. Usually an Equivalent line, Exponential function, Hyperbolic function, Polynom or the XY Lookup Table should be used. Characteristics based on XY-Data pairs allow simple access to measured curves for the switching behavior of the components.

To control the MOS, a logical signal is applied. This signal can originate from any SIMPLORER system variable.

Control Signal (Base current)	Component State
> 0	ON
≤ 0	OFF

$$v(t) = f(i(t), [\text{control signal}])$$

Dialog Settings	
⊙	«Characteristic Type» The MOS type with predefined characteristics is selected. In the list, select a characteristic.
▾	«Type» In the list, select the equivalent line, exponential function, or XY Data Pairs and define the corresponding characteristic parameters. These are Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Equivalent Line (MOSEQUL):
The MOS characteristic is defined by an Equivalent line. You have to enter a value for the Forward voltage, Bulk resistance and Reverse resistance.

V_{limit} stands for the intersection of the forward and reverse characteristic and is calculated by the program itself.

$$I = \frac{v(t) - V_f}{R_b} \quad \text{for} \quad v(t) \geq V_{limit}$$

$$I = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < V_{limit}$$

Exponential Function (MOSEXP):
The MOS characteristic is defined by an Exponential function. You have to enter values for the Saturation current, Thermal voltage and Reverse resistance.

$$i(t) = I_s \cdot \left(e^{\frac{v(t)}{V_t}} - 1\right) \quad \text{for} \quad v(t) \geq 0$$

$$i(t) = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < 0$$

XY Data Pairs (MOSXY)
The MOS characteristic is defined by a data table. You can define the table within the characteristic dialog directly or load an existing file. The data pairs are displayed in the lookup table and used for the characteristic. You can modify the data when necessary. See also “Separate Component Characteristic” on page 297.

⊙ **«Element Name»**
The characteristic of the MOS is defined by a characteristic component. Characteristic parameter type. Enter the name in the text box (e.g. `XY1.VAL`, `EXP.VAL`) or use the pin to connect the characteristic component. The internal used diode type is **MOS**.

☑ ☐ **«Use pin»**
If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.

«Control Signal»
Defines the MOS control signal. Control parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.

☑ ☐ **«Use pin»**
If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic; MOS	CH	real
Control Signal; All components	CTRL	real
Forward Voltage [V]; MOSEQUL	VF	real
Bulk Resistance [Ω]; MOSEQUL	RB	real
Reverse Resistance [Ω]; MOSEQUL/MOSEXP	RR	real
Saturation Current [A]; MOSEXP	ISAT	real
Thermal Voltage [V]; MOSEXP	VT	real
Characteristic; MOSXY	FILE	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Conductivity [S]	G	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Drain	D	electrical
Source	S	electrical

Bipolar Junction Transistor

>>Basics>Circuit>Semiconductors System Level

The System level BJT is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the selected characteristic and the applied control signal.

The BJT types of predefined characteristics defined within the component dialog guarantee a high calculation speed. In contrast to the predefined characteristic types (equivalent line, exponential function) you can use each other separate characteristic from the *Tools* folder or an appropriate quantity of a component. Usually an Equivalent line, Exponential function, Hyperbolic function, Polynom or the XY Lookup Table should be used. Characteristics based on XY-Data pairs allow simple access to measured curves for the switching behavior of the components.

To control the BJT, a logical signal is applied. This signal can originate from any SIMPLORER system variable.

Control Signal (Base current)	Component State
> 0	ON
≤ 0	OFF

$$v(t) = f(i(t), [\text{control signal}])$$

Dialog Settings	
⦿	«Characteristic Type» The BJT type with predefined characteristics is selected. In the list, select a characteristic.
▼	«Type» In the list, select the equivalent line, exponential function, or XY Data Pairs and define the corresponding characteristic parameters. These are Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

	Equivalent Line (BJTEQUL): The BJT characteristic is defined by an Equivalent line. You have to enter a value for the Forward voltage, Bulk resistance and Reverse resistance. V_{limit} stands for the intersection of the forward and reverse characteristic and is calculated by the program itself. $$I = \frac{v(t) - V_f}{R_b} \quad \text{for} \quad v(t) \geq V_{limit}$$ $$I = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < V_{limit}$$
	Exponential Function (BJTEXP): The BJT characteristic is defined by an Exponential function. You have to enter values for the Saturation current, Thermal voltage and Reverse resistance. $$i(t) = I_s \cdot \left(e^{\frac{v(t)}{V_t}} - 1\right) \quad \text{for} \quad v(t) \geq 0$$ $$i(t) = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < 0$$
	XY Data Pairs (BJTXY) The BJT characteristic is defined by a data table. You can define the table within the characteristic dialog directly or load an existing file. The data pairs are displayed in the lookup table and used for the characteristic. You can modify the data when necessary. See also “Separate Component Characteristic” on page 297.
⊙	**«Element Name»** The characteristic of the BJT is defined by a characteristic component. Characteristic parameter type. Enter the name in the text box (e.g. `XY1.VAL`, `EXP.VAL`) or use the pin to connect the characteristic component. The internal used type is BJT.
☑ ☐	**«Use pin»** If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.
	«Control Signal» Defines the BJT control signal. Control parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	**«Use pin»** If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic; BJT	CH	real
Control Signal; All components	CTRL	real
Forward Voltage [V]; BJTEQUL	VF	real
Bulk Resistance [Ω]; BJTEQUL	RB	real
Reverse Resistance [Ω]; BJTEQUL/BJTEXP	RR	real
Saturation Current [A]; BJTEXP	ISAT	real
Thermal Voltage [V]; BJTEXP	VT	real
Characteristic; BJTXY	FILE	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Conductivity [S]	G	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Collector	C	electrical
Emitter	E	electrical

GTO-Thyristor

>>Basics>Circuit>Semiconductors System Level

The System level GTO is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the selected characteristic and the applied control signal. The GTO types of predefined characteristics defined within the component dialog guarantee a high calculation speed. In contrast to the predefined characteristic types (equivalent line, exponential function) you can use each other separate characteristic from the *Tools* folder or an appropriate quantity of a component. Usually an Equivalent line, Exponential function, Hyperbolic function, Polynom or the XY Lookup Table should be used. Characteristics based on XY-Data pairs allow simple access to measured curves for the switching behavior of the components. To control the GTO-Thyristor, a logical signal is applied. This signal can originate from any SIMPLORER system variable.

Control signal (ignition current)	**Component State**
> 0	ON, if v(t) > VF
< 0	OFF
= 0	No change (the GTO turns off, when the current through the component becomes ≤ 0)

$$v(t) = f(i(t), [\text{control signal}])$$

Dialog Settings	
⊙	«Characteristic Type» The GTO type with predefined characteristics is selected. In the list, select a characteristic.
▼	«Type» In the list, select the equivalent line, exponential function, or XY Data Pairs and define the corresponding characteristic parameters. These are Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

	Equivalent Line (GTOEQUL): The GTO characteristic is defined by an Equivalent line. You have to enter a value for the Forward voltage, Bulk resistance and Reverse resistance. V_{limit} stands for the intersection of the forward and reverse characteristic and is calculated by the program itself. $$I = \frac{v(t) - V_f}{R_b} \quad \text{for} \quad v(t) \geq V_{limit}$$ $$I = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < V_{limit}$$ **Exponential Function (GTOEXP):** The GTO characteristic is defined by an Exponential function. You have to enter values for the Saturation current, Thermal voltage and Reverse resistance. $$i(t) = I_s \cdot \left(e^{\frac{v(t)}{V_t}} - 1\right) \quad \text{for} \quad v(t) \geq 0$$ $$i(t) = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < 0$$ **XY Data Pairs (GTOXY)** The GTO characteristic is defined by a data table. You can define the table within the characteristic dialog directly or load an existing file. The data pairs are displayed in the lookup table and used for the characteristic. You can modify the data when necessary. See also “Separate Component Characteristic” on page 297.
⊙	**«Element Name»** The characteristic of the GTO is defined by a characteristic component. Characteristic parameter type. Enter the name in the text box (e.g. `XY1.VAL`, `EXP.VAL`) or use the pin to connect the characteristic component. The internal used type is **GTO.**
☑ ☐	**«Use pin»** If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.
	«Control Signal» Defines the GTO control signal. Control parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	**«Use pin»** If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic; GTO	CH	real
Control Signal; All components	CTRL	real
Forward Voltage [V]; GTOEQUL	VF	real
Bulk Resistance [Ω]; GTOEQUL	RB	real
Reverse Resistance [Ω]; GTOEQUL/GTOEXP	RR	real
Saturation Current [A]; GTOEXP	ISAT	real
Thermal Voltage [V]; GTOEXP	VT	real
Characteristic; GTOXY	FILE	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Conductivity [S]	G	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Anode	A	electrical
Cathode	C	electrical

Thyristor

>>Basics>Circuit>Semiconductors System Level

The System level Thyristor is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the selected characteristic and the applied control signal.

The thyristor types of predefined characteristics defined within the component dialog guarantee a high calculation speed. In contrast to the predefined characteristic types (equivalent line, exponential function) you can use each other separate characteristic from the *Tools* folder or an appropriate quantity of a component. Usually an Equivalent line, Exponential function, Hyperbolic function, Polynom, or the XY Lookup Table should be used. Characteristics based on XY-Data pairs allow simple access to measured curves for the switching behavior of the components. To control the thyristor, a logical signal is applied. This signal can originate from any SIMPLORER system variable.

Control Signal (ignition current)	Component state
> 0	ON, if v(t) > VF
≤ 0	no change (the thyristor turns off, when the current through the component becomes ≤ 0)

$$v(t) = f(i(t), [\text{control signal}])$$

Dialog Settings	
⊙	«Characteristic Type» The thyristor type with predefined characteristics is selected. In the list, select a characteristic.
▼	«Type» In the list, select the equivalent line, exponential function, or XY Data Pairs and define the corresponding characteristic parameters. These are Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

	Equivalent Line (THEQUL): The thyristor characteristic is defined by an Equivalent line. You have to enter a value for the Forward voltage, Bulk resistance and Reverse resistance. V_{limit} stands for the intersection of the forward and reverse characteristic and is calculated by the program itself. $I = \frac{v(t) - V_f}{R_b}$ for $v(t) \geq V_{limit}$ $I = \frac{v(t)}{R_r}$ for $v(t) < V_{limit}$ **Exponential Function (THEXP):** The thyristor characteristic is defined by an Exponential function. You have to enter values for the Saturation current, Thermal voltage and Reverse resistance. $i(t) = I_s \cdot \left(e^{\frac{v(t)}{V_t}} - 1\right)$ for $v(t) \geq 0$ $i(t) = \frac{v(t)}{R_r}$ for $v(t) < 0$ **XY Data Pairs (THXY)** The thyristor characteristic is defined by a data table. You can define the table within the characteristic dialog directly or load an existing file. The data pairs are displayed in the look-up table and used for the characteristic. You can modify the data when necessary. See also "Separate Component Characteristic" on page 297.
⊙	**«Element Name»** The characteristic of the thyristor is defined by a characteristic component. Characteristic parameter type. Enter the name in the text box (e.g. `XY1.VAL`, `EXP.VAL`) or use the pin to connect the characteristic component. The internal used type is **TH.**
☑ ☐	**«Use pin»** If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.
	«Control Signal» Defines the thyristor control signal. Control parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	**«Use pin»** If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic; TH	CH	real
Control Signal; All components	CTRL	real
Forward Voltage [V]; THEQUL	VF	real
Bulk Resistance [Ω]; THEQUL	RB	real
Reverse Resistance [Ω]; THEQUL/THEXP	RR	real
Saturation Current [A]; THEXP	ISAT	real
Thermal Voltage [V]; THEXP	VT	real
Characteristic; THXY	FILE	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Conductivity [S]	G	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Anode	A	electrical
Cathode	C	electrical

TRIAC

>>Basics>Circuit>Semiconductors System Level

The System level TRIAC is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the selected characteristic and the applied control signal.

The TRIAC types of predefined characteristics defined within the component dialog guarantee a high calculation speed. In contrast to the predefined characteristic types (equivalent line, exponential function) you can use each other separate characteristic from the *Tools* folder or an appropriate quantity of a component. Usually an Equivalent line, Exponential function, Hyperbolic function, Polynom, or the XY Lookup Table should be used. Characteristics based on XY-Data pairs allow simple access to measured curves for the switching behavior of the components. This signal can originate from any SIMPLORER system variable.

Control signal (ignition current)	Component state
> 0	ON, if $v(t) > VF$
≤ 0	No change

$$v(t) = f(i(t), [\text{control signal}])$$

Dialog Settings	
⊙	«Characteristic Type» The TRIAC type with predefined characteristics is selected. In the list, select a characteristic.
▼	«Type» In the list, select the equivalent line, exponential function, or XY Data Pairs and define the corresponding characteristic parameters. These are Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Equivalent Line (TRIACEQUL):
The TRIAC characteristic is defined by an Equivalent line. You have to enter a value for the Forward voltage, Bulk resistance and Reverse resistance.

V_{limit} stands for the intersection of the forward and reverse characteristic and is calculated by the program itself.

$$I = \frac{v(t) - V_f}{R_b} \quad \text{for} \quad v(t) \geq V_{limit}$$

$$I = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < V_{limit}$$

Exponential Function (TRIACEXP):
The TRIAC characteristic is defined by an Exponential function. You have to enter values for the Saturation current, Thermal voltage and Reverse resistance.

$$i(t) = I_s \cdot \left(e^{\frac{v(t)}{V_t}} - 1 \right) \quad \text{for} \quad v(t) \geq 0$$

$$i(t) = \frac{v(t)}{R_r} \quad \text{for} \quad v(t) < 0$$

XY Data Pairs (TRIACXY)
The thyristor characteristic is defined by a data table. You can define the table within the characteristic dialog directly or load an existing file. The data pairs are displayed in the look-up table and used for the characteristic. You can modify the data when necessary. See also “Separate Component Characteristic” on page 297.

⊙ **«Element Name»**
The characteristic of the TRIAC is defined by a characteristic component. Characteristic parameter type. Enter the name in the text box (e.g. `XY1.VAL`, `EXP.VAL`) or use the pin to connect the characteristic component. The internal used type is **TRIAC.**

☑ ☐ **«Use pin»**
If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.

«Control Signal»
Defines the TRIAC control signal. Control parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.

☑ ☐ **«Use pin»**
If the box is checked, the name is defined by the quantity connected to the pin, otherwise the name in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic [/]; TRIAC	CH	real
Control Signal [/]; All components	CTRL	real
Forward Voltage [V]; TRIACEQUL	VF	real
Bulk Resistance [Ω]; TRIACEQUL	RB	real
Reverse Resistance [Ω]; TRIACEQUL/TRIACEXP	RR	real
Saturation Current [A]; TRIACEXP	ISAT	real
Thermal Voltage [V]; TRIACEXP	VT	real
Characteristic; TRIACXY	FILE	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Conductivity [S]	G	real
Derivative of Voltage [V/s]	dV	real
Derivative of Current [A/s]	dI	real

Component Nodes

Description	Node Name	Nature
Pin 1	N1	electrical
Pin 2	N2	electrical

4.5 Semiconductors Device Level

- Diode
- IGBT
- FET
- BJT
- OPV
- GTO
- Thyristor

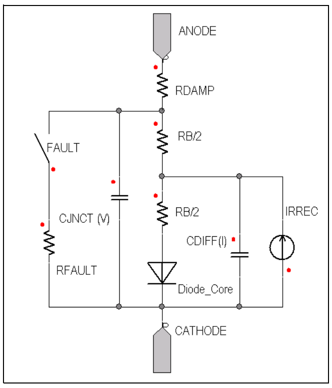
Dynamic Diode Model

>>Basics>Circuit>Semiconductors Device Level
The diode model is a modular model with definable simulation levels. Different simulation depths can be selected for the electrical and thermal behavior of the model.

You can define three simulation levels for both the electrical and thermal behavior. Each level combination has a certain set of parameters. The values of them can be defined in the model dialog. The component outputs are true of all types of the diode models. They are listed at the end of the model description.

Dialog Settings	
▼	«Electrical Behavior Level» In the list, select the Electrical Behavior Level—0,1, 2, or 3.
↓↑	«Electrical Parameters» Define the electrical diode parameters according to the selected electrical behavior level. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
▼	«Thermal Behavior Level» In the list, select the Thermal Behavior Level —0,1, or 2.
↓↑	«Thermal Parameters» Define the thermal diode parameters according to the selected thermal behavior level. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Electrical Model



The model uses an electrical equivalent circuit.There are three simulation levels regarding the electrical behavior.

TYPE_DYN=0
Only the static behavior is calculated. The charges at the capacitances are ignored, no switching behavior.

TYPE_DYN=1
In addition to the static behavior, charging and discharging of junction and diffusion capacitance are calculated.

TYPE_DYN=2 and DYN=3
If this level is selected, an additional current source to model the reverse recovery behavior is used.

Electrical Behavior Level, Type DYN=0

Static diode current:

$$I = IS \cdot \left(e^{\frac{V - I \cdot RB}{M \cdot V_T}} - 1 \right) \qquad V_T = \frac{k \cdot (TEMP\_JNCT + 273K)}{q}$$

k = Boltzmann constant (1.381E-23) $\qquad$ q = Elementary charge (1.602E-19)

Saturation current:

The saturation current is calculated from the given saturation current at reference temperature TEMP0 and the actual junction temperature TEMP_JNCT.

$$IS = ISAT0 \cdot \left(\frac{TEMP\_JNCT + 273K}{TEMP0 + 273K} \right)^3 \cdot e^{\left(\frac{q \cdot VGAP}{k \cdot (TEMP\_JNCT + 273K)} \cdot \frac{TEMP\_JNCT - TEMP0}{TEMP0 + 273K} \right)}$$

Component Parameters

Description [Unit]	Parameter Name	Data Type
Saturation Current at Tc0 [A]	ISAT0	real
Ideality Factor [/]	M0	real
Linear Temperature Coefficient of M [/]	ALPHA_M	real
Bulk Resistance [Ω]	RB0	real
Exponentail Temperature Coefficient [/]	ALPHA_RB	real
Bandgap Voltage, Si 1.1 [V]	VGAP	real
Reference Temperature [°C]	TEMP0	real
Breakthrough Voltage [V]	VBREAK	real
Breakthrough Current [V]	IBREAK	real
Breakthrough Junction Temperature [°C]	TEMPBREAK	real
Resistance Anode-Cathode after Fault	RFAULT	real

Electrical Behavior Level, Type DYN=1

Junction capacitance:

There is a distincion between the calculation of depletion and enhancement capacitance behavior. The curves remain differentiable even at the transition from one region to the other. The transition happens if the junction voltage is less than 0 V.

If the junction voltage is greater than 0 V, the following equation is used:

$$C_{JNCT} = C0_{JNCT} \cdot \left(2 - e^{-\left(\frac{V_{JUNCT} \cdot ALPHA \cdot (1 - DELTA)}{VDIFF} \right)} \right)$$

If the junction capacitance is negative (depletion region), the following equation is used:

$$C_{JNCT} = C0_{JNCT} \cdot \left(DELTA + \frac{1 - DELTA}{\left(\frac{1 - V_{JUNCT}}{VDIFF} \right)^{ALPHA}} \right)$$

4

Diffusion capacitance:

$$\mathrm{CDIFF} = \mathrm{TAU} \cdot \frac{d(I_{JNCT})}{d(V_{JNCT})} = \mathrm{TAU} \cdot \frac{(I_{JNCT} + \mathrm{IS})}{M \cdot V_T}$$

To damp the oscillations that occur in the system, additional damping resistances are placed in the diode model. The value of the damping resistance depends on the parameter DAMPING, the value of the parasitic inductance and the values of the internal capacitances.

Damping resistance:

$$\mathrm{RDAMP} = \mathrm{DAMPING} \cdot \sqrt{\frac{L}{C(V)}}$$

Component Parameters (+ Parameters of TYPE DYN=0)

Description [Unit]	Parameter Name	Data Type
Nominal Blocking Voltage [V]	VNOM	real
Nominal Current [A]	INOM	real
Total Junction Capacitance at 0V [F]	C0_JNCT	real
Diffusion Potential [V]	VDIFF_JNCT	real
Capacitance Exponent [/]	ALPHA_JNCT	real
Influence of constant Capacitance [0…1] [/]	DELTA_JNCT	real
Effective Lifetime [s]	TAU	real
Exponential Temperature Coefficient of TAU [/]	ALPHA_TAU	real
Linear Current Coefficient of TAU [/]	KAPPA_TAU	real
Linear Voltage Coefficient of TAU [/]	SIGMA_TAU	real
Parasitic Inductance [H]	L	real
Damping Factor [/]	DAMPING	real

Electrical Behavior Level, Type DYN=2,3

The integral of the reverse recovery current is determined by the parameter TAU. Form factors define the curve of the reverse recovery current. The Electrical Behavior Level to 2 and 3 represent two different characteristics. In both the reverse recovery is separated into five sections. The maximum current Irrmax, and the time ts, when Irrmax is reached, are calculated from the reverse recovery charge and the form factors.

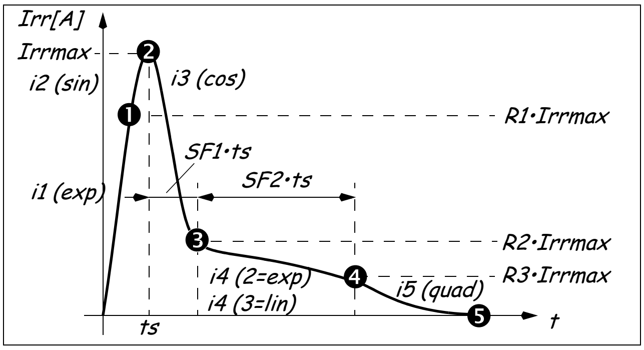
1: From zero crossing of diode current until the level R1*Irrmax is reached. The current follows an exponential function with the time constant TAU.

2: The current follows a sine wave function until Irrmax is reached.

3: The current follows a cosine function. This section lasts for SF1*ts. At its end the current is at the level R2*Irrmax.

4: If the Electrical Behavior Level is set to '2' the current follows an exponential function. For level 3 a linear function is used. Section 4 ends SF2*ts after Irrmax was reached. The current at the end of section 4 is R3*Irrmax.

5: A quadratic function is used until the current reaches zero again.



Component Parameters (+ Parameters of TYPE DYN=0 and TYPE DYN=1)

Description [Unit]	Parameter Name	Data Type
Reverse Recovery Form Factor 1, 2, and 3	R1/R2/R3	real
Exponential Temperature Coefficient of R2	ALPHA_TAU	real
Linear Current Coefficient of R2	KAPPA_TAU	real
Linear Voltage Coefficient of R2	SIGMA_TAU	real
Reverse Recovery Soft Factor 1 and 2	SF1/SF2	real

Temperature, Voltage, and Current Dependencies

The model parameters may depend on junction temperature, terminal voltage and current. For the representation of these dependencies three groups of parameters were introduced:

- Coefficients for temperature dependencies
  The names of these parameters start with ALPHA
- Coefficients for voltage dependencies
  The names of these parameters start with SIGMA
- Coefficients for current dependencies
  The names of these parameters start with KAPPA

Two different functions are used to model the dependencies. The original function is y=f(x)

The dependency on an additional value z (temperature, voltage, current) is expressed by the coefficient in the following way (m stand for ALPHA, SIGMA, or KAPPA):

Linear Dependency:

$$y_m = \left(1 + m \cdot \frac{z - z_0}{z_0}\right) f(x)$$

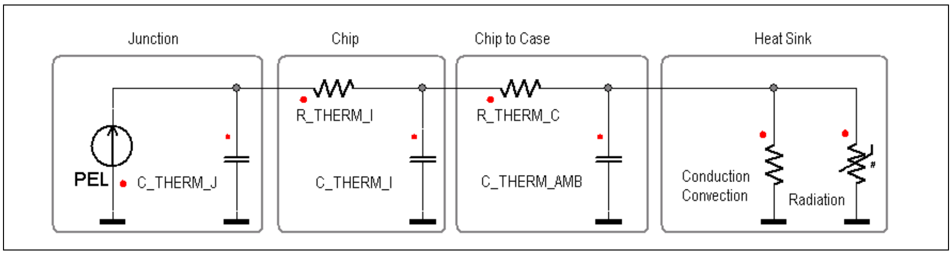
Exponential Dependency:

$$y_m = \left(\frac{z}{z_0}\right)^m f(x)$$

Temperature dependency can be linear or exponential while voltage and current dependencies are always linear. Some parameters have more than one coefficient. In this case the individual terms are multiplied. The reference values are always TEMP0, VNOM and INOM respectively.

Thermal Semiconductor Model

The structure of the thermal model is shown below.



If the built-in model is not sufficient for the simulation purpose, it is possible to use the model output PEL to control a current source in a more complex thermal network. The simulated junction temperature from this network can be fed into the electrical model. In this case TYPE_THERM = 0 has to be set.

The thermal behavior can be defined with the parameter TYP_THERM. There are three levels of the thermal behavior:

TYPE_THERM = 0
The junction temperature is set to the ambient temperature (Parameter 'TEMPAMB'). All temperature dependent semiconductor parameters are re-calculated according to this temperature.

TYPE_THERM=1
The junction and case temperatures are influenced by the internal losses and the thermal resistances. All thermal capacitances are ignored (static thermal model).

TYPE_THERM=2
The temperatures are calculated using a dynamic thermal model taking into account all thermal resistances and capacitances.

Component Parameters TYPE THERM=0

Description [Unit]	Parameter Name	Data Type
Ambient Temperature [°C]	TEMPAMB	real

Component Parameters TYPE THERM=1 (+ Parameters of TYPE THERM=0)

Description [Unit]	Parameter Name	Data Type
Thermal Intrinsic Resistance [K/W]	R_THERM_I	real
Thermal Resistance Chip to Case [K/W]	R_THERM_C	real

Component Parameters TYPE THERM=2 (+ Parameters of TYPE THERM=0 and THERM=1)

Description [Unit]	Parameter Name	Data Type
Junction Temperature at Simulation Start [°C]	TEMPJNCT0	real
Thermal Junction Capacitance [Ws/K]	C_THERM_J	real
Thermal Intrinsic Capacitance [Ws/K]	C_THERM_I	real
Thermal Ambient Capacitance [Ws/K]	C_THERM_AMB	real

Convection Number (>=1m) [W/cm^{2}/K]	ALPHA_CONV	real
Convection Area [cm^2]	A_CONV	real
Thermal Radiation Constant (>=5.7p) [W/cm^2/K4]	SIGMA_RAD	real
Thermal Radiation Area [cm^2]	A_RAD	real
Thermal Conductivity to Ambient [W/K]	G_COND_AMB	real

Component Outputs for all Diode Types

Description [Unit]	Parameter Name	Data Type
Current [A]	I	real
Anode-Cathode-Voltage [V]	V	real
Junction Capacitance [F]	CJNCT	real
Diffusion Charge [As]	QDIFF	real
Diffusion Capacitance [F]	CDIFF	real
Reverse Recovery Charge [As]	QRREC	real
Junction Temperature [°C]	TEMPJNCT	real
Chip Temperature [°C]	TEMPINTR	real
Case Temperature [°C]	TEMPC	real
Diode Losses [W]	PEL	real
Power Transfer by Conduction [W]	PCOND	real
Power Transfer by Convection [W]	PCONV	real
Power Transfer by Radiation [W]	PRAD	real
Losses of One Switching Cycle [Ws]	ESWITCH	real
Average Losses of One Switching Cycle [W]	PSWITCH	real
Total Losses During Simulation [Ws]	ETOT	real
Total Average Losses During Simulation [W]	PTOT	real
Flag Over voltage [/]	FAULT_V	real
Flag Over current [/]	FAULT_I	real
Flag Over temperature [/]	FAULT_TEMP	real

Component Nodes

Description	Node Name	Nature
Anode	ANODE	electrical
Cathode	CATHODE	electrical

Dynamic IGBT Model

>>Basics>Circuit>Semiconductors Device Level

The IGBT model is a modular model with definable simulation levels. Different simulation depths can be selected for the electrical and thermal behavior of the model. For the models with a free wheeling diode (FWD) the simulation depth of the diode model can be set separately.

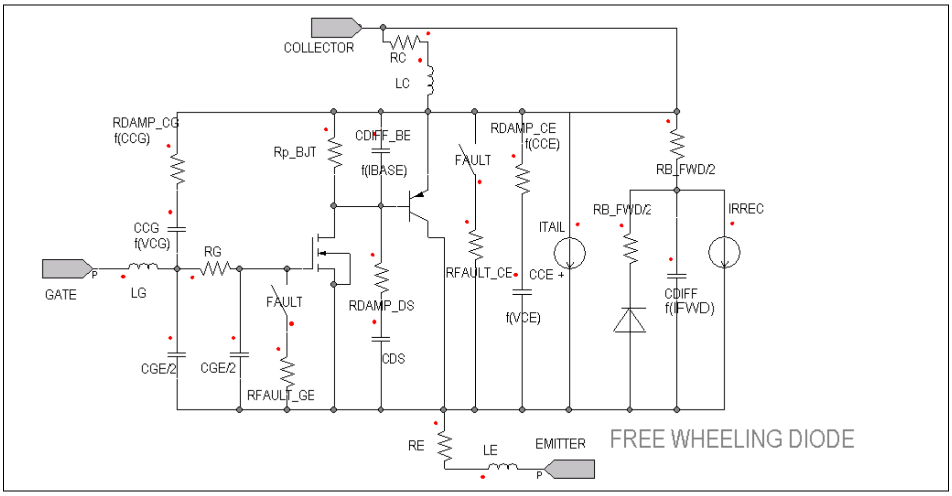
The IGBT is a combination of an MOSFET and a bipolar junction transistor. The internal MOSFET is similar to the single MOSFET model. Since the internal BJT is always in saturation, only its current gain BN0 is available as parameter of the IGBT model. The internal model of the free wheeling diode (FWD) is similar to that of the single diode.

You can define five simulation levels for the electrical and three simulation levels for the thermal behavior. Each level combination has a certain set of parameters. The values of them can be defined in the model dialog. The component outputs are true of all types of the IGBT models. They are listed at the end of the model description. See also “Thermal Semiconductor Model” on page 126 and “Temperature, Voltage, and Current Dependencies” on page 125.

Dialog Settings	
▾	«Electrical Behavior Level» In the list, select the Electrical Behavior Level —0,1, 2, 3, or 4.
↓↑	«Electrical Parameters» Define the electrical IGBT parameters according to the selected electrical behavior level. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
▾	«Thermal Behavior Level» In the list, select the Thermal Behavior Level —0,1, or 2.
↓↑	«Thermal Parameters» Define the thermal IGBT parameters according to the selected thermal behavior level. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
▾	«Freewheeling Diode Behavior Level» In the list, select the Thermal Behavior Level —1, or 2.
	«Freewheeling Diode» Define the freewheeling diode parameters. They are effective for all behavior levels. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Electrical Model

The model uses an electrical equivalent circuit as shown in the figure below.



There are five simulation levels regarding the electrical behavior. The level is set at the parameter TYPE_DYN:

TYPE_DYN=0
Only the static behavior is calculated. The charges at the capacitances are ignored, no switching behavior.

TYPE_DYN=1
In addition to the static behavior, charging and discharging of the junction and diffusion capacitance are calculated. Values of parasitic capacitances at the terminals can be defined.

TYPE_DYN=2
If this level is selected, the model is expanded by a current source that models the tail current.

TYPE_DYN=3
The tail current is calculated considering the junction temperature and the influence of the collector current on the time constant of the tail current.

TYPE_DYN=4
At this level the dependency of the tail current on the collector-emitter voltage is taking into account in addition to all other effects modeled in the previous levels.

Please note that the value of the voltage influencing the tail current is taken from the previous switching-off. That is why the level does not give proper results for fault simulation and should be used only for investigations of periodic switching processes.

Electrical Behavior Level, Type DYN=0

For the calculation of the static FET current a distinction is made between the linear region and the pinch-off region. The so-called saturation voltage is given with

$$V_{sat} = A\_FET \cdot (V_{GS} - V_P)^{M\_FET}$$

The transition happens when the drain current satisfies the following equation

$$I_{sat} = \frac{k}{2} \cdot (V_{GS} - V_P)^{N\_FET}$$

Within the linear region the following equation is used

$$I_D = I_{sat} \cdot (1 + KLM \cdot V_{DS}) \cdot \left(2 - \frac{V_{DS}}{V_{sat}}\right) \cdot \frac{V_{DS}}{V_{sat}}$$

and for pinch-off applies

$$I_D = I_{sat} \cdot (1 + KLM \cdot V_{DS})$$

The static IGBT current is calculated from

$$IC = I_D \cdot B_N$$

The transistor constant k, the pinch-off voltage VP and the current gain BN are temperature dependent. Their values at the nominal temperature TEMP0 are parameters of the model. During the simulation their actual values are calculated taking into account the actual value of the junction temperature taken from the thermal model.

The static current of the FWD (if enabled) and the current into the base of the BJT follows the formula:

$$I = I_S \cdot \left(e^{\frac{V_{AK}}{M \cdot V_T}} - 1 \right) \qquad V_T = \frac{k \cdot (TEMP\_JNCT + 273K)}{q}$$

k = Boltzmann constant (1.381E-23) $\qquad$ q = Elementary charge (1.602E-19)

The saturation current IS is temperature dependent and is calculated using the parameter ISAT0 and the actual junction temperature imported from the thermal model.

The model has a built-in fault detection. During the simulation the collector current, the voltage drop from gate to emitter and from collector to emitter as well as the junction temperature are observed. If their limitations are exceeded the model behavior changes. The switches controlled by the FAULT flags are closed and the respective fault resistances determine the model characteristic. If an over voltage occurs across Gate-Emitter, only RFAULT_GE comes into effect. All other faults effect the RFAULT_CE.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Saturation Factor of internal FET	A_FET	real
Exponential Cofficient of A_FET	ALPHA_AFET	real
Saturation Exponent of internal FET	M_FET	real
Exponential Cofficient of M_FET	ALPHA_MFET	real
Exponent of Transfer Characteristic of FET	N_FET	real
Exponential Cofficient of N_FET	ALPHA_NET	real
Transistor Constant of FET at TEMP0	K0	real
Exponential Temperature Cofficient of K	ALPHA_K	real
Pinch-off-Voltage of FET at TEMP0 [V]	VP0	real
Exponential Temperature Cofficient of VP	ALPHA_VP	real
Channel Lenght Modulation Factor of FET [1/V]	KLM	real
Band Gap Voltage of FET [V]	VGAP	real
Current Gain of BJT at TEMP0	BN0	real
Saturation Current of BJT Base at TEMP0 [A]	ISAT0_BJT	real
Exponential Temperature Cofficient of ISAT	ALPHA_ISAT_BJT	real
Ideality Factor of BJT Base at TEMP0	M0_BJT	real
Linear Temperature Cofficient of M_BJT [Ω]	ALPHA_M_BJT	real
Bulk Resistance of BJT Base [Ω]	RB0	real
BJT Base Shut [Ω]	RP_BJT	real
Collector Connector Resistance [Ω]	RC	real
Emitter Connector Resistance [Ω]	RE	real
Internal Gate Resistance for Switching On [Ω]	RGON	real
Exponential Temperature Cofficient of RGON	ALPHA_RGON	real
Linear Current Cofficient of RGON	KAPPA_RGON	real
Linear Voltage Cofficient of RGON	SIGMA_RGON	real

Internal Gate Resistance for Switching OFF [Ω]	ROFF	real
Exponential Temperature Cofficient of RGOFF	ALPHA_RGOFF	real
Linear Current Cofficient of RGOFF	KAPPA_RGOFF	real
Linear Voltage Cofficient of RGOFF	SIGMA_RGOFF	real
Reference Temperature [°C]	TEMP0	real
Breakthrough Collector-Emitter Voltage [V]	VBREAK_CE	real
Breakthrough Gate-Emitter Voltage [V]	VBREAK_GE	real
Breakthrough Collector Current [A]	IBREAK	real
Breakthrough Junction Temperature [°C]	TEMPBREAK	real
Breakthrough Junction Temperature [°C]	TEMPBREAK	real
Collector-Emitter Resistance after Fault [Ω]	RFAULT_CE	real
Gate-Emitter Resistance after Fault [Ω]	RFAULT_GE	real

Electrical Behavior Level, Type DYN=1

All capacitance between the terminals are modeled the same way. A distinction is made between a region where a depletion capacitance is calculated and a region with enhancement capacitance behavior. The curves remain differentiable even at the transition from one region to the other. The transition happens if the difference of a defined fraction of the diffusion and the voltage across the junction equals zero according to

$$\mathrm{SHIFT} \cdot \mathrm{VDIFF} - V_{\mathrm{JNCT}} = V^*_{\mathrm{JNCT}} = 0$$

For positive values of V^*_{JNCT} (enhancement region) the following formula is used to calculate the capacitance

$$C(V^*_{\mathrm{JNCT}}) = \mathrm{C0} \cdot \left(1 + (\mathrm{BETA} - 1) \cdot \left(1 - e^{-\frac{V^*_{\mathrm{JNCT}} \cdot \mathrm{ALPHA} \cdot (1 - \mathrm{DELTA})}{(\mathrm{BETA} - 1) \cdot \mathrm{VDIFF}}}\right)\right)$$

At negative values of V^*_{JNCT} (depletion region) the following formula applies

$$C(V^*_{\mathrm{JNCT}}) = \mathrm{C0} \cdot \left(\mathrm{DELTA} + \frac{1 - \mathrm{DELTA}}{\left(1 - \frac{V^*_{\mathrm{JNCT}}}{\mathrm{VDIFF}}\right)^{\mathrm{APHA}}}\right)$$

The diffusion capacitances at the base-emitter path of the BJT and at the FWD are modeled using the same approach:

$$C_{\mathrm{diff}} = \mathrm{TAU} \cdot \frac{i(t) + \mathrm{ISAT}}{M \cdot V_T}$$

To damp the oscillations that may occur in the system, additional damping resistances are placed in the model. The values of the damping resistances depend on the parameter DAMPING, the values of the parasitic inductances and the values of the internal capacitances.

$$R_{\mathrm{DAMP}} = \mathrm{DAMPING} \cdot \sqrt{\frac{L}{C(V)}}$$

Component Parameters (+ Parameters of TYPE DYN=0)

Description [Unit]	Parameter Name	Data Type
Nominal Collector-Emitter Voltage [V]	VNOM	real
Nominal Collector Current [A]	INOM	real
Gate-Emitter Capacitance [F]	C0_GE	real
Exponential Temperature Cofficient of CO_GE	ALPHA_C0_GE	real
Linear Current Cofficient of RGON of CO_GE	KAPPA_C0_GE	real
Linear Voltage Cofficient of RGON of CO_GE	SIGMA_C0_GE	real
Gate-Emitter Diffusion Potential [V]	VDIFF_GE	real
Gate-Emitter Voltage Shift [V]	VSHIFT_GE	real
Gate-Emitter Capacitance Exponent	ALPHA_GE	real
Gate-Emitter Peak Factor	BETA_GE	real
Exponential Temperature Cofficient of BETA_GE	ALPHA_BETA_GE	real
Linear Current Cofficient of BETA_GEE	KAPPA_BETA_GE	real
Linear Voltage Cofficient of BETA_GE	SIGMA_BETA_GE	real
Influence of constant Capacitance at Gate Emitter	DELTA_GE	real
Collector-Gate Reference Capacitance [F]	CO_CG	real
Diffusion Potential of Collector-Gate Capacitance [V]	VDIFF_CG	real
Voltage Shift of Collector-Gate Capacitance [V]	VSHIFT_CG	real
Capacitance Exponent Collector-Gate	ALPHA_CG	real
Peak Factor of Collector-Gate Capacitance	BETA_CG	real
Influence of constant Capacitance at Collector-Gate	DELTA_CG	real
Collector-Emitter Reference Capacitance [F]	CO_CE	real
Diffusion Potential of Collector-Emitter Capacitance [F]	VDIFF_CE	real
Voltage Shift of Collector-Emitter Capacitance [V]	VSHIFT_CE	real
Capacitance Exponent Collector-Emitter	ALPHA_CE	real
Peak Factor of Collector-Emitter Capacitance	BETA_CE	real
Influence of constant Capacitance at Collector-Emitter	DELTA_CE	real
Relative Value of Internal Drain-Source Capacitance	CR_DS	real
Exponential Temperature Coefficient of CR_DS	ALPHA_CR_DS	real
Linear Current Coefficient of CR_DS	KAPPA_CR_DS	real
Linear Voltage Coefficient of CR_DS	SIGMA_CR_DS	real
Carrier Lifetime at Base-Emitter Junction [s]	TAU_BE	real
Exponential Temperature Coefficient of TAU_BE	ALPHA_TAU_BE	real
Linear Current Coefficient of TAU_BE	KAPPA_TAU_BE	real
Linear Voltage Coefficient of TAU_BE	SIGMA_TAU_BE	real
Collector Connector Inductance [H]	LC	real
Emitter Connector Inductance [H]	LG	real
Source Connector Inductance [H]	LE	real
Damping Factor	DAMPING	real

Electrical Behavior Level, Type DYN=2

At this level the model is expanded by a current source that models the tail current.

Component Parameters (+ Parameters of TYPE DYN=0 and TYPE DYN=1)

Description [Unit]	Parameter Name	Data Type
Time Constant of Tail Current [s]	TAU_TAIL	real
Exponential Temperature Coefficient of TAU_TAIL	ALPHA_TAU_TAIL	real
Linear Current Coefficient of TAU_TAIL	KAPPA_TAU_TAIL	real
Linear Voltage Coefficient of TAU_TAIL	SIGMA_TAU_TAIL	real
Start of Tail Current relative to Maximum Current	DELTA_TAIL	real
Exponential Temperature Coefficient of DELTA_TAIL	ALPHA_DELTA_TAIL	real
Linear Current Coefficient of DELTA_TAIL	KAPPA_DELTA_TAIL	real
Linear Voltage Coefficient of DELTA_TAIL	SIGMA_DELTA_TAIL	real

Electrical Behavior Level, Type DYN=3

The tail current is calculated considering the junction temperature and the influence of the collector current on the time constant of the tail current. For parameters see level 2.

Electrical Behavior Level, Type DYN=4

At this level the dependency of the tail current on the collector-emitter voltage is taking into account in addition to all other effects modeled in the previous levels. There are no additional parameters.

Please note: the value of the voltage influencing the tail current is taken from the previous switching-off. Therefore the level does not give proper results for fault simulation and should be used only for investigations of periodic switching processes. For parameters see level 2.

Freewheeling Diode Model

The freewheeling diode model is the same as the single diode model without junction capacitance. All equations of it apply. Via TYPE_FWD the behavior of the diode model can be set.

TYPE_FWD=0	No FWD
TYPE_FWD=1	Static FWD
TYPE_FWD=2 or 3	Dynamic FWD with reverse recovery

Component Parameters FWD Model

Description [Unit]	Parameter Name	Data Type
Ideality Factor of FWD at TEMP0 [/]; static parameter	M0_FWD	real
Saturation Current of FWD at TEMP0 [A]; static parameter	ISAT0_FWD	real
Bulk Resistance of FWD at TEMP0 [Ω]; static parameter	RB0_FWD	real
Linear Temperature Coefficient of M_FWD; static parameter	ALPHA_M_FWD	real
Linear Temperature Coefficient of ISAT_FWD; static parameter	ALPHA_ISAT_FWD	real

Linear Temperature Coefficient of RB_FWD	ALPHA_RB_FWD	real
Effective Lifetime of FWD	TAU_FWD	real
Exponential Temperature Coefficient of TAU_FWD	ALPHA_TAU_FWD	real
Linear Current Coefficient of TAU_FWD	KAPPA_TAU_FWD	real
Linear Voltage Coefficient of TAU_FWD	SIGMA_TAU_FWD	real
Reverse Recovery Forma Factor 1	R1_FWD	real
Reverse Recovery Forma Factor 2	R2_FWD	real
Exponential Temperature Coefficient of R2_FWD	ALPHA_R2_FWD	real
Linear Current Coefficient of TAU_FWD	KAPPA_R2_FWD	real
Linear Voltage Coefficient of TAU_FWD	SIGMA_R2_FWD	real
Reverse Recovery Forma Factor 3	R3_FWD	real
Soft Factor 1 of FWD	SF1_FWD	real
Soft Factor 2 of FWD	SF2_FWD	real
Exponential Temperature Coefficient of SF2_FWD	ALPHA_SF2_FWD	real
Linear Current Coefficient of SF2_FWD	KAPPA_SF2_FWD	real
Linear Voltage Coefficient of SF2_FWD	SIGMA_SF2_FWD	real

Component Outputs for all IGBT Types

Description [Unit]	Parameter Name	Data Type
Gate-Emitter Voltage [V]	VGE	real
Base-Emitter Voltage of BJT [V]	VBE	real
Collector-Emitter Voltage [V]	VCE	real
Current through Collector Connector [A]	IC	real
Current through Gate Connector [A]	IG	real
Current through FWD [A]	I_FWD	real
Pinch-off Voltage [V]	VP	real
Transistor Constant	K	real
Current Gain of BJT	BN	real
Gate-Emitter Capacitance [F]	CGE	real
Collector-Gate Capacitance [F]	CCG	real
Collector-Emitter Capacitance [F]	CCE	real
Gate-Source Capacitance [F]	CDS	real
Capacitance at FWD [F]	CFWD	real
Tail Current Charge [As]	Q_TAIL	real
Junction Temperature [°C]	TEMPJNCT	real
Chip Temperature [°C]	TEMPCHIP	real
Case Temperature [°C]	TEMPCASE	real
Total Component Losses [W]	PEL	real
Power Transfer by Conduction [W]	PCOND	real
Power Transfer by Convection [W]	PCONV	real
Power Transfer by Radiation [W]	PRAD	real
Losses of One Switching Cycle [Ws]	ESWITCH	real

Average Losses of One Switching Cycle [W]	PSWITCH	real
Total Losses During Simulation [Ws]	ETOT	real
Total Average Losses During Simulation [W]	PTOT	real
Flag Collector-Emitter-Over voltage	FAULT_VCE	real
Flag Gate-Emitter-Over voltage	FAULT_VGE	real
Flag Over current	FAULT_I	real
Flag Over temperature	FAULT_TEMP	real

Component Nodes

Description	Node Name	Nature
Collector	COLLECTOR	electrical
Emitter	EMITTER	electrical
Auxiliary Emitter	AUX_EMITTER	electrical
Gate	GATE	electrical

Dynamic FET Model

>>Basics>Circuit>Semiconductors Device Level

The FET model is a modular model with definable simulation levels. Different simulation depths can be selected for the electrical and thermal behavior of the model.

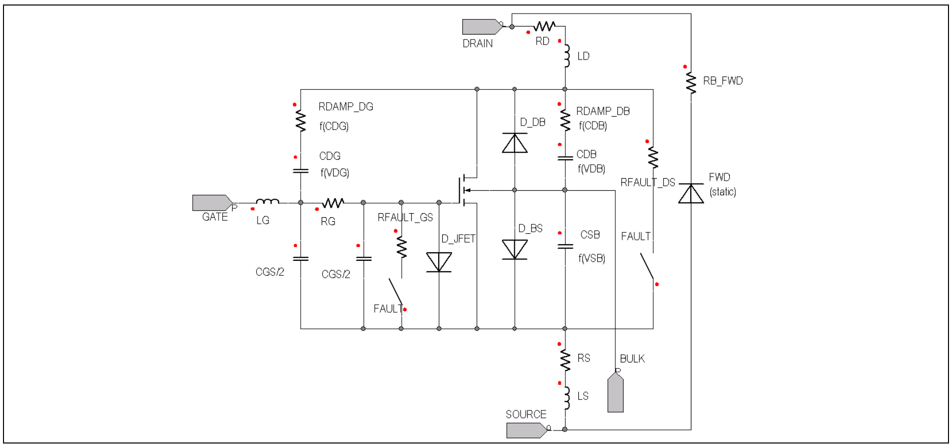
At normal operation of the device the diodes D_DB and D_BS are blocking. They have default parameters that cannot be changed by the user.

You can define two simulation levels for the electrical and three simulation levels for the thermal behavior. Each level combination has a certain set of parameters. The values of them can be defined in the model dialog. The component outputs are true of all types of the FET models. They are listed at the end of the model description. See also “Thermal Semiconductor Model” on page 126 and “Temperature, Voltage, and Current Dependencies” on page 125.

Dialog Settings	
▼	«Electrical Behavior Level» In the list, select the Electrical Behavior Level —0, or 1.
↓↑	«Electrical Parameters» Define the electrical FET parameters according to the selected electrical behavior level. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
▼	«Thermal Behavior Level» In the list, select the Thermal Behavior Level —0,1, or 2.
↓↑	«Thermal Parameters» Define the thermal FET parameters according to the selected thermal behavior level. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
	«Freewheeling Diode» Define the freewheeling diode parameters. They are effective for all behavior levels. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Electrical Model

The model uses an electrical equivalent circuit as shown in the figure below.



There are two simulation levels regarding the electrical behavior. The level is set at the parameter TYPE_DYN:

TYPE_DYN=0

Only the static behavior is calculated. The charges at the capacitances are ignored, no switching behavior.

TYPE_DYN=1

In addition to the static behavior, charging and discharging of the junction and diffusion capacitance are calculated. Values of parasitic capacitances at the terminals can be defined.

Electrical Behavior Level, Type DYN=0

For the calculation of the static FET current a distinction is made between the linear region and the pinch-off region. The so-called saturation voltage is given with

$$V_{sat} = A\_FET \cdot (VGS - V_P)^{M\_FET}$$

The transition happens when the drain current satisfies the following equation

$$I_{sat} = \frac{k}{2} \cdot (VGS - V_P)^{N\_FET}$$

Within the linear region the following equation is used

$$ID = I_{sat} \cdot (1 + KLM \cdot VDS) \cdot \left(2 - \frac{VDS}{V_{sat}}\right) \cdot \frac{VDS}{V_{sat}}$$

and for pinch-off applies

$$ID = I_{sat} \cdot (1 + KLM \cdot VDS)$$

The model has a built-in fault detection. During the simulation the drain current, the voltage drop from gate to source and from drain to source as well as the junction temperature are observed. If their limitations are exceeded the model behavior changes. The switches controlled by the FAULT flags are closed and the respective fault resistances determine the model characteristic. If an over voltage across Gate-Source occurs, only RFAULT_GS comes into effect. All other faults effect the RFAULT_DS.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Transistor Constant at TEMP0 [A/V²]	K0	real
Pinch-off-Voltage at TEMP0 [V]	VP0	real
Bulk Control Factor	BCF	real
Bulk Doping [1/cm³]	NBULK	real
Channel Doping [1/cm³]	NCH	real
Channel Lenght Modulation Factor	KLM	real
Saturation Factor	A_FET	real
Saturation Exponent	M_FET	real
Exponent of Transfer Characteristic	N_FET	real
Ideality Factor of Gate Diode; only JFET	M0_GATE	real
Saturation Diode of Gate Diode; only JFET	ISAT0_GATE	real
Bulk Resistance of Gate Diode; only JFET	RB0_GATE	real
Reference Temperature [°C]	TEMP0	real
Band Gap Voltage [V]	VGAP	real
Linear Temperature Coefficient of VP	ALPHA_VP	real
Exponential Temperature Coefficient of Transistor Constant	ALPHA_K	real
Exponential Temperature Coefficient of A_FET	ALPHA_AFET	real
Exponential Temperature Coefficient of M_FET	ALPHA_MFET	real
Exponential Temperature Coefficient of N_FET	ALPHA_NFET	real
Breakthrough G-S-Voltage [V]	VBREAK_DS	real
Breakthrough Drain-Source-Voltage [V]	VBREAK_GS	real
Breakthrough Drain Current [A]	IBREAK	real
Breakthrough Junction Temperature [°C]	TEMPBREAK	real
Drain-Source-Resistance after Fault [Ω]	RFAULT_DS	real
Gate-Source-Resistance after Fault [Ω]	RFAULT_GS	real
Drain Connector Resisance	RD	real
Gate Connector Resistance	RG	real
Source Connector Resistance	RS	real

Electrical Behavior Level, Type DYN=1

Apart from the constant gate-source-capacitance, all capacitances are modeled in the same way. A distinction is made between a region where a depletion capacitance is calculated and a region with enhancement capacitance behavior. The curves remain differentiable even at the transition from one region to the other. The transition happens if the difference of a defined fraction of the diffusion and the voltage across the junction equals zero according to:

$$\mathrm{SHIFT} \cdot \mathrm{VDIFF} - V_{JNCT} = V^*_{JNCT} = 0$$

For positive values of V^*_{JNCT} (enhancement region) the following formula is used to calculate the capacitance:

$$C(V^*_{JNCT}) = C0 \cdot \left(1 + (BETA - 1) \cdot \left(1 - e^{-\frac{V^*_{JNCT} \cdot ALPHA \cdot (1 - DELTA)}{(BETA - 1) \cdot VDIFF}}\right)\right)$$

At negative values of V^*_{JNCT} (depletion region) the following formula applies:

$$C(V^*_{JNCT}) = C0 \cdot \left(DELTA + \frac{1 - DELTA}{\left(1 - \frac{V^*_{JNCT}}{VDIFF}\right)^{APHA}}\right)$$

To damp the oscillations that may occur in the system, additional damping resistances are placed in the model. The values of the damping resistances depend on the parameter DAMPING, the values of the parasitic inductances and the values of the internal capacitances.

$$R_{DAMP} = DAMPING \cdot \sqrt{\frac{L}{C(V)}}$$

Component Parameters (+ Parameters of TYPE DYN=0)

Description [Unit]	Parameter Name	Data Type
Gate-Source Capacitance [F]	C0_GS	real
Drain-Gate Reference Capacitance [F]	C0_DG	real
Diffusion Potential of Drain-Gate Capacitance [V]	VDIFF_DG	real
Voltage Shift of Drain-Gate Capacitance [V]	VSHIFT_DG	real
Capacitance Exponent Drain-Gate	ALPHA_DG	real
Peak Factor of Drain-Gate Capacitance	BETA_DG	real
Influence of constant. Capacitance at Drain-Gate	DELTA_DG	real
Drain Bulk Reference Capacitance [F]	C0_DB	real
Diffusion Potential of Drain-Bulk Capacitance [V]	VDIFF_DB	real
Voltage Shift of Drain-Bulk Capacitance [V]	VSHIFT_DB	real
Capacitance Exponent Drain-Bulk	ALPHA_DB	real
Peak Factor of Drain-Bulk Capacitance	BETA_DB	real
Influence of constant. Capacitance at Drain-Bulk	DELTA_DB	real
Source-Bulk Reference Capacitance [F]	C0_SB	real
Diffusion Potential of SB Capacitance [V]	VDIFF_SB	real
Voltage Shift of Source-Bulk Capacitance [V]	VSHIFT_SB	real
Capacitance Exponent Source-Bulk	ALPHA_SB	real
Peak Factor of Source-Bulk Capacitance	BETA_SB	real
Influence of constant. Capacitance at Source-Bulk	DELTA_SB	real
Damping Factor	DAMPING	real
Drain Connector Inductance [H]	LD	real
Gate Connector Inductance [H]	LG	real
Source Connector Inductance [H]	LS	real
Nominal Drain-Source Voltage [V]	VNOM	real
Nominal Drain Current [I]	INOM	real

Freewheeling Diode Model

The FWD model is a static one only. The static current follows the formula:

$$I = I_{SAT,FWD} \cdot \left(e^{\frac{-VDS - I \cdot RB\_FWD}{M \cdot V_T}} - 1 \right) \qquad V_T = \frac{k \cdot (TEMP\_JNCT + 273K)}{q}$$

k = Boltzmann constant (1.381E-23) $\qquad$ q = Elementary charge (1.602E-19)

Component Parameters FWD Model

Description [Unit]	Parameter Name	Data Type
Ideality Factor of FWD	M_FWD	real
Saturation Current of FWD at TEMP0 [A]	ISAT0_FWD	real
Bulk Resistance of FWD [Ω]	RB_FWD	real
Linear Temperature Coefficient of M_FET	ALPHA_M_FWD	real
Exponential Temperature Coefficient of N_FET	ALPHA_ISAT_FWD	real
Linear Temperature Coefficient of M_FWD	ALPHA_RB_FWD	real

Component Outputs for all FET Types

Description [Unit]	Parameter Name	Data Type
Internal Gate-Source-Voltage [V]	VGS	real
Internal Gate-Drain-Voltage [V]	VGD	real
Internal Drain-Source-Voltage [V]	VDS	real
Internal Bulk-Source-Voltage [V]	VBS	real
Bulk-Drain-Voltage [V]	VBD	real
Current through Drain Connector [A]	ID	real
Current through Gate Connector [A]	IG	real
Gate-Source-Capacitance [F]	CGS	real
Drain-Gate-Capacitance [F]	CDG	real
Drain-Bulk-Capacitance [F]	CDB	real
Source-Bulk-Capacitance [F]	CSB	real
Junction Temperature [°C]	TEMPJNCT	real
Chip Temperature [°C]	TEMPINTR	real
Case Temperature [°C]	TEMPC	real
Total Component Losses [W]	PEL	real
Power Transfer by Conduction [W]	PCOND	real
Power Transfer by Convection [W]	PCONV	real
Power Transfer by Radiation [W]	PRAD	real
Losses of One Switching Cycle [Ws]	ESWITCH	real
Average Losses of One Switching Cycle [W]	PSWITCH	real
Total Losses During Simulation [Ws]	ETOT	real
Total Average Losses During Simulation [W]	PTOT	real

Flag Drain-Source-Over voltage	FAULT_VDS	real
Flag Gate-Source-Over voltage	FAULT_VGS	real
Flag Over current	FAULT_I	real
Flag Over temperature	FAULT_TEMP	real

Component Nodes

Description	Node Name	Nature
GATE	GATE	electrical
SOURCE	SOURCE	electrical
DRAIN	DRAIN	electrical
BULK	BULK	electrical

Dynamic BJT Model

>>Basics>Circuit>Semiconductors Device Level

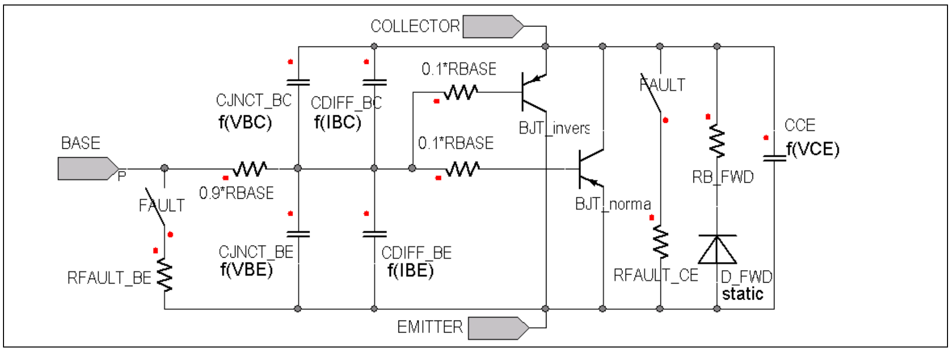
The BJT model is a modular model with definable simulation levels. Different simulation depths can be selected for the electrical and thermal behavior of the model.

You can define two simulation levels for the electrical and three simulation levels for the thermal behavior. Each level combination has a certain set of parameters. The values of them can be defined in the model dialog. The component outputs are true of all types of the BJT models. They are listed at the end of the model description. See also “Thermal Semiconductor Model” on page 126 and “Temperature, Voltage, and Current Dependencies” on page 125..

Dialog Settings	
▼	«Electrical Behavior Level» In the list, select the Electrical Behavior Level —0, or 1.
	«Electrical Parameters» Define the electrical BJT parameters according to the selected electrical behavior level. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
▼	«Thermal Behavior Level» In the list, select the Thermal Behavior Level —0,1, or 2.
	«Thermal Parameters» Define the thermal BJT parameters according to the selected thermal behavior level. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
	«Freewheeling Diode» Define the freewheeling diode parameters. They are effective for all behavior levels. Look their meaning up in the parameter table below. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Electrical Model

The model uses an electrical equivalent circuit as shown in the figure below.



There are two simulation levels regarding the electrical behavior. The level is set at the parameter TYPE_DYN:

TYPE_DYN=0
Only the static behavior is calculated. The charges at the capacitances are ignored, no switching behavior.

TYPE_DYN=1
In addition to the static behavior, charging and discharging of the junction and diffusion capacitances are calculated.

Electrical Behavior Level, Type DYN=0

The static transistor current is determined by the two BJT models for normal and inverse operation. In the normal operation mode, where the internal base-emitter voltage and the collector-emitter voltage are above 0 V, the current gain values for the BJT models are calculated from:

$$B_N = BN0 \cdot \left(1 + \frac{VCE}{VEARLY}\right) \qquad B_I = BI0$$

In the inverse mode applies:

$$B_N = BN0 \qquad B_I = BI0 \cdot \left(1 - \frac{VCE}{VEARLY}\right)$$

The static collector current is calculated from the base currents of the BJT models:

$$IC = B_N \cdot I_{BN} - I_{BI}(1 + B_I)$$

The base currents of the BJT models are calculated from their base-emitter voltages:

$$I_{B*} = I_{SAT} \cdot \left(e^{\frac{VBE*}{M\_BJT \cdot V_T}} - 1\right) \qquad V_T = \frac{k \cdot (TEMP\_JNCT + 273K)}{q}$$

$$k = \text{Boltzmann constant (1.381E-23)} \qquad q = \text{Elementary charge (1.602E-19)}$$

The saturation voltage of the BJT model for normal mode follows from:

$$I_{SAT,N} = ISAT0\_BJT \cdot \left(\frac{TEMP\_JNCT + 273K}{TEMP0 + 273K}\right)^3 \cdot e^{\left(\frac{VGAP}{V_T} \cdot \frac{TEMP\_JNCT - TEMP0}{TEMP0 + 273K}\right)}$$

The saturation voltage of the BJT model for inverse operation follows from:

$$I_{SAT,I} = I_{SAT,N} \cdot \frac{B_N}{B_I}$$

The model has a built-in fault detection. During the simulation the collector current and the base current, the voltage drop across base-emitter and collector-emitter as well as the junction temperature are observed. If their limitations are exceeded the model behavior changes. The switches controlled by the FAULT flags are closed and the respective fault resistances determine the model characteristic. Faults at the base do only effect RFAULT_BE. All other faults effect the RFAULT_CE.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Normal Current Gain	BN0	real
Exponential Temperature Coefficient of BN	ALPHA_BN	real
Inverse Current Gain	BI0	real
Exponential Temperature Coefficient of BI	ALPHA_BI	real
Early Voltage [V]	VEARLY	real
Exponential Temperature Coefficient of VEARLY	ALPHA_VEARLY	real
Ideality Factor of Base Junction	M0	real
Exponential Temperature Coefficient of Base Junction Ideality Factor	ALPHA_M	real
Saturation Current of Base Junction [A]	ISAT0	real
Exponential Temperature Coefficient of Base Junction Saturation Current [A]	ALPHA_ISAT	real
Bulk Resitance of Base Junction[Ω]	RB0	real
Exponential Temperature Coefficient of Base Junction Bulk Resistance	ALPHA_RB	real
Band Gap Voltage [V]	VGAP	real
Collector Connector Resistance [Ω]	RC	real
Base Connector Resistance [Ω]	RB	real
Emitter Connector Resistance [Ω]	RE	real
Reference Temperature [°C]	TEMP0	real
Breakthrough Collector-Emitter Voltage [V]	VBREAK_CE	real
Breakthrough Base-Emitter Voltage [V]	VBREAK_BE	real
Breakthrough Collector Current [A]	IBREAK_C	real
Breakthrough Base Current [A]	IBREAK_B	real
Breakthrough Junction Temperature [°C]	TEMPBREAK	real
Collector-Emitter Resistance after Fault [Ω]	RFAULT_CE	real
Base-Emitter Resistance after Fault [Ω]	RFAULT_BE	real

Electrical Behavior Level, Type DYN=1

Junction capacitance:

There is a distincion between the calculation of depletion and enhancement capacitance behavior. The curves remain differentiable even at the transition from one region to the other. The transition happens if the junction voltage is less than 0 V.

If the junction voltage is greater than 0 V, the following equation is used:

$$C_{JNCT} = C0_{JNCT} \cdot \left(2 - e^{-\left(\frac{V_{JUNCT} \cdot ALPHA \cdot (1 - DELTA)}{VDIFF}\right)} \right)$$

If the junction capacitance is negative (depletion region), the following equation is used:

$$C_{JNCT} = C0_{JNCT} \cdot \left(DELTA + \frac{1 - DELTA}{\left(\frac{1 - V_{JUNCT}}{VDIFF}\right)^{ALPHA}} \right)$$

Diffusion capacitances (base-emitter, base-collector):

$$CDIFF\_BE = TAU\_BE \cdot \frac{I_{BN}}{M\_BJT \cdot V_T} \qquad CDIFF\_BC = TAU\_BC \cdot \frac{I_{BI}}{M\_BJT \cdot V_T}$$

Component Parameters (+ Parameters of TYPE DYN=0)

Description [Unit]	Parameter Name	Data Type
Nominal Voltage [V]	VNOM	real
Nominal Current [A]	INOM	real
Base-Emitter Reference Capacitance [F]	C0_BE	real
Diffusion Potential of Base-Emitter Capacitance [V]	VDIFF_BE	real
Capacitance Exponent Base-Emitter	ALPHA_BE	real
Influence of constant capacitance at Base-Emitter	DELTA_BE	real
Exponential Temperature Coefficient of Base-Emitter Depletion Capacitance	ALPHA_CBE	real
Base-Collector Reference Capacitance [F]	C0_BC	real
Diffusion Potential of Base-Collector Capacitance [V]	VDIFF_BC	real
Capacitance Exponent Base-Collector	ALPHA_BC	real
Influence of constant capacitance at Base-Collector	DELTA_BC	real
Collector-Emitter Reference Capacitance [F]	C0_CE	real
Diffusion Potential of Collector-Emitter Capacitance [V]	VDIFF_CE	real
Capacitance Exponent Collector-Emitter	ALPHA_CE	real
Influence of constant capacitance at Collector-Emitter	DELTA_CE	real
Carrier Lifetime at Base-Emitter Junction [s]	TAU_BE	real
Temperature Coefficient of Base-Emitter Carrier Lifetime	ALPHA_TAU_BE	real
Carrier Lifetime at Base-Collector Junction [s]	TAU_BC	real
Temperature Coefficient of Base-Collector Carrier Lifetime	ALPHA_TAU_BC	real
Collector Connector Inductance [H]	LC	real
Base Connector Inductance [H]	LB	real
Emitter Connector Inductance [H]	LE	real

4

FWD Model

The FWD model is a static one only. The static current follows the formula:

$$I = I_{SAT,\,FWD} \cdot \left(e^{\frac{-VDS - I \cdot RB\_FWD}{M \cdot V_T}} - 1 \right) \qquad V_T = \frac{k \cdot (TEMP\_JNCT + 273K)}{q}$$

k = Boltzmann constant (1.381E-23) $\qquad$ q = Elementary charge (1.602E-19)

Component Parameters FWD Model

Description [Unit]	Parameter Name	Data Type
Ideality Factor of FWD	M_FWD	real
Saturation Current of FWD at TEMP0 [A]	ISAT0_FWD	real
Bulk Resistance of FWD [Ω]	RB_FWD	real

Component Outputs for all BJT Types

Description [Unit]	Parameter Name	Data Type
Base-Emitter Voltage [V]	VBE	real
Base-Collector Voltage [V]	VBC	real
Collector-Emitter-Voltage [V]	VCE	real
Current through Collector Connector [A]	IC	real
Current through Base Connector [A]	IB	real
Base-Emitter Capacitance [F]	CBE	real
Base-Collector Capacitance [F]	CBC	real
Collector-Emitter Capacitance [F]	CCE	real
Junction Temperature [°C]	TEMPJNCT	real
Chip Temperature [°C]	TEMPINTR	real
Case Temperature [°C]	TEMPC	real
Total Component Losses [W]	PEL	real
Power Transfer by Conduction [W]	PCOND	real
Power Transfer by Convection [W]	PCONV	real
Power Transfer by Radiation [W]	PRAD	real
Losses of One Switching Cycle [Ws]	ESWITCH	real
Average Losses of One Switching Cycle [W]	PSWITCH	real
Total Losses During Simulation [Ws]	ETOT	real
Total Average Losses During Simulation [W]	PTOT	real
Flag Collector-Emitter Over voltage	FAULT_VCE	real
Flag Base-Emitter Over voltage	FAULT_VBE	real
Flag Collector Over current	FAULT_IC	real
Flag Base Over current	FAULT_IB	real
Flag Over temperature	FAULT_TEMP	real

Component Nodes

Description	Node Name	Nature
Base	BASE	electrical
Emitter	EMITTER	electrical
Collector	COLLECTOR	electrical

Dynamic OPV Model

>>Basics>Circuit>Semiconductors Device Level

The model represents a differential-voltage amplifiers with or without universal frequency compensation for digital and analog applications. The operational amplifier is a general amplifier with differential input. It is used for controlling purposes, analog computation, filters, video technics, digital technics, ADC/DAC and power amplifiers in the analog or switched case.

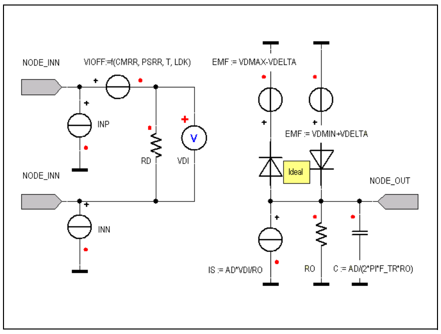
The power supply of the operational amplifier is not modeled as an electrical node. Instead of that, the supply voltage and, if necessary, the noise on it, can be entered as parameters of the mode. The frequency characteristic of the operational amplifier determined only by the parameter F_TR which is the transit frequency of the model. At this frequency, the gain of the operational amplifier equals '1'. Please note: The thermal behavior not implemented! Uncompensated frequency behavior with oscillation tendency (phase shift < -90°), this means instability of the gain cannot be modelled.

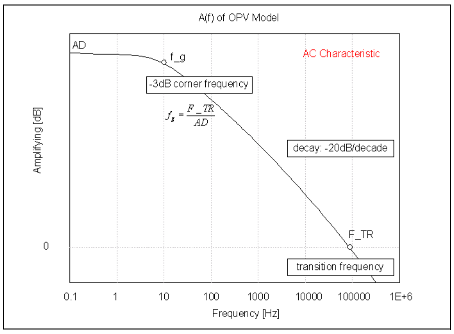
Dialog Settings
«Parameter List 1 to define an OPV» Define the values of OpAmp parameters in the table. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Look the meaning up in the parameter table below.
«Parameter List 2 to define an OPV» Define the values of OpAmp parameters in the table. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Look the meaning up in the parameter table below.

Internal Voltage Source:

$$V_{OFF} = VIOFF + D_G \cdot VCM + D_P \cdot VNSUPPLY + COEFF_{VIOFF} \cdot (TEMP - TEMP0) + D_t$$

$$D_G = 10^{(-CMRR/20)} \qquad D_P = 10^{(-PSRR/20)} \qquad D_t = LDF \cdot \sin\left(\frac{2\pi t}{86.4 \cdot 10^3}\right)$$





Component Parameters

Description [Unit]	Parameter Name	Data Type
Differential Gain [V/V]	AD	real
Maximum Output Voltage, only OPV without supply pin [V]	VOMAX	real
Minimum Output Voltage, only OPV without supply pin [V]	VOMIN	real
Input-Offset-Voltage [V]	VIOFF0	real
Input Quiescent Current at Negative Input [A]	INN	real
Input Quiescent Current at Positive Input [A]	INP	real
Differential Input Resistance [Ω]	RD	real
Output Resistance [Ω]	RO	real
Common Mode Rejection Ratio [dB]	CMRR	real
Power Supply Rejection [dB]	PSRR	real
Temperature [°C]	TEMP	real
Reference Temperature [°C]	TEMP0	real
Temperature coefficient of VIOFF [V/K]	COEFF_VIOFF	real
Long Term Drift Coefficient [V/h]	LDK	real
Min. Difference to Supply Voltage, OPV without supply [V]	VDELTA	real
Noise at Supply Voltage, OPV without supply [V]	VNSUPPLY	real
Transit Frequency [Hz]	F_TR	real
Slew Rate, only OPV with supply pin [Vs]	SR	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Differential Input Voltage [V]	VD	real
Common Input Voltage [V]	VCM	real
Ideal Output Voltage without Limitation [V]	VOUT_IDEAL	real

Component Nodes

Description	Node Name	Nature
Non-Inverting Input	NINV	electrical
Inverting Input	INV	electrical
Output	OUT	electrical

Dynamic GTO Model

>>Basics>Circuit>Semiconductors Device Level

The model is a macro implementation of a dynamic GTO model. The static behavior of the model (normal mode and reverse operation) is determined by static diode models and current controlled current sources. The dynamic behavior is determined by nonlinear capacitances (junction and diffusion capacitances).

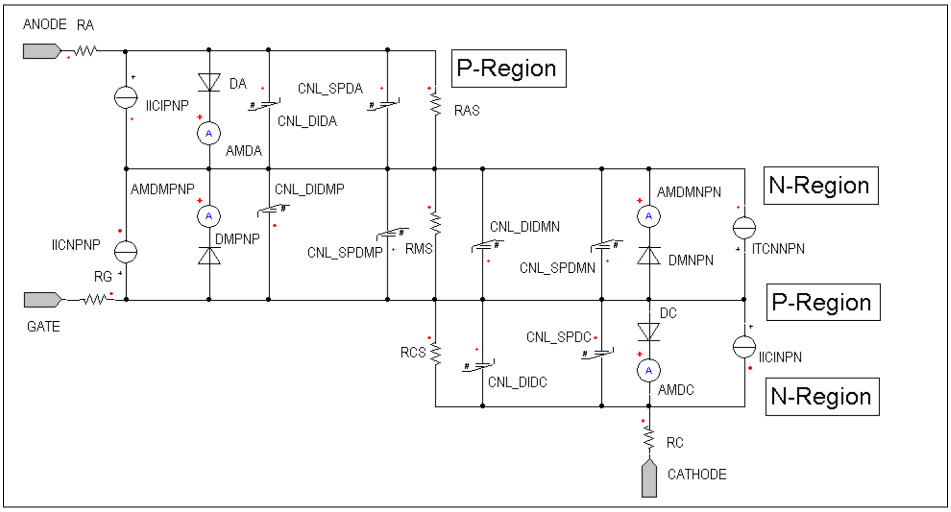
All diode models use an exponential characteristic with the same value of saturation current defined by the model parameter IS (see figure below). The emission coefficients NF and NR effect the thermal voltages of the internal diode models. NF effects the diodes at the Anode and Cathode; NR is used for the diodes DMPNP and DMNPN which model the reverse operation mode of the GTO.

For all current amplification factors in common mode constant values are used. The current amplification of ITCNNPN is adjustable by the parameter K. Recommended values for K are in the range of 1 to 5.

The model does not consider the influence of critical rates of change in voltage and current in the outer circuit on the switching behavior of the GTO. Due to the short time constants of the systems, a small value of the minimum step size (usually some nanoseconds) should be set. In the following the influences of the parameters on the model behavior are listed:

Element	Modification	Effect
CNL_DIDA	Increase of TT1	Increase of current rise time Increase of peak reverse current
CNL_SPDA	Increase of CJOI	Increase of Gate-controlled delay time
CNL_DIDMP, CNL_DIDMN	Increase of TT2	Increase of critical hold-off interval
CNL_SPDMP, CNL_SPDMN	Increase of CJO2	Decrease of gate-controlled delay time
CNL_DIDC	Increase of TT3	Increase of current rise time
CNL_SPDC	Increase of CJO3	Increase of gate-controlled delay time
ICNNPN	Increase of K	Increase of gate trigger current

Dialog Settings
«Static Parameters» Define the static GTO parameters in the table. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Look the meaning up in the parameter table below.
«Dynamic Parameters» Define the dynamic GTO parameters in the table. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Look the meaning up in the parameter table below.



Component Parameters

Description [Unit]	Parameter Name	Data Type
Saturation Current [A], static parameter	IS	real
Forward Emission Coefficient, static parameter	NF	real
Reverse Emission Coefficient, static parameter	NR	real
Bulk Resistance Anode [Ω], static parameter	RA	real
Bulk Resistance Gate [Ω], static parameter	RG	real
Bulk Resistance Cathode [Ω], static parameter	RC	real
Amplification Factor, static parameter	K	real
Carrier lifetime [s], dynamic parameter	TT1	real
Carrier lifetime [s], dynamic parameter	TT2	real
Carrier lifetime [s], dynamic parameter	TT3	real
Junction Capacitance [F], dynamic parameter	CJO1	real
Junction Capacitance [F], dynamic parameter	CJO2	real
Junction Capacitance [F], dynamic parameter	CJO3	real

Component Nodes

Description	Node Name	Nature
Anode Pin	ANODE	electrical
Gate Pin	GATE	electrical
Cathode Pin	CATHODE	electrical

Dynamic Thyristor Model

>>Basics>Circuit>Semiconductors Device Level

The model is a macro implementation of a dynamic thyristor model. The static behavior of the model (normal mode and reverse operation) is determined by static diode models and current controlled current sources. The dynamic behavior is determined by nonlinear capacitances (junction and diffusion capacitances).

All diode models use an exponential characteristic with the same value of saturation current defined by the model parameter IS (see figure below). The emission coefficients NF and NR effect the thermal voltages of the internal diode models. NF effects the diodes at the Anode and Cathode; NR is used for the diodes DMPNP and DMNPN which model the reverse operation mode of the thyristor.

For all current amplification factors in common mode constant values are used. The current amplification of ITCNNPN is adjustable by the parameter K. Recommended values for K are in the range of 0,01 to 0,5.

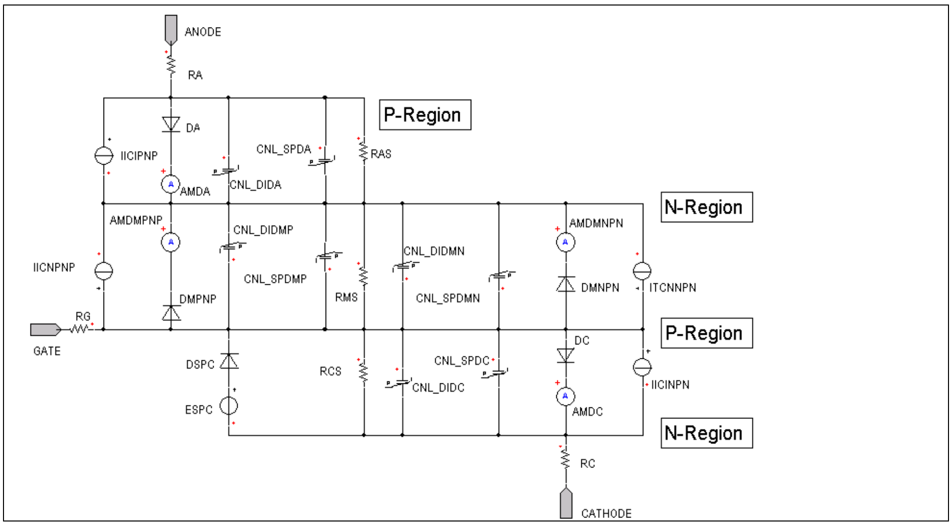
The model does not consider the influence of critical rates of change in voltage and current in the outer circuit on the switching behavior of the thyristor. Due to the short time constants of the systems, a small value of the minimum step size (usually some nanoseconds) should be set. In the following the influences of the parameters on the model behavior are listed:

Element	Modification	Effect
CNL_DIDA	Increase of TT1	Increase of current rise time Increase of peak reverse current
CNL_SPDA	Increase of CJOI	Increase of Gate-controlled delay time
CNL_DIDMP, CNL_DIDMN	Increase of TT2	Increase of critical hold-off interval
CNL_SPDMP, CNL_SPDMN	Increase of CJO2	Decrease of gate-controlled delay time
CNL_DIDC	Increase of TT3	Increase of current rise time
CNL_SPDC	Increase of CJO3	Increase of gate-controlled delay time
ICNNPN	Increase of K	Increase of igniting current

Typical parameter values and characteristic values for thyristors of different ratings are:

Nominal current	25A	100A	1500A
IS	1*E-14	1*E-14	1*E-3
NF	1,5	1,5	1,5
NR	1	1	1
TT1	100*E-9	300*E-9	300*E-9
TT2	25*E-9	200*E-9	200*E-9
TT3	10*E-9	50*E-9	1500*E-9
CJ01	10*E-9	1*E-9	10*E-9
CJ02	1*E-9	1*E-9	1*E-9
CJ03	40*E-9	40*E-9	100*E-9
RA	1*E-3	1*E-3	1*E-3
RG	5*E-3	5*E-3	7,5
RC	1*E-3	10*E-6	10*E-6
K	0.01	0.03	0.25
Igniting current [mA]	60	80	400
Holding current [mA]	55	110	800
Igniting voltage [V]	1.4	1.4	3.6
Forward voltage [V]	1.9	1.9	2.3
Turn-on time [ms]	1.1	1.1	1.1
Current rise time [ms]	0.15	0.45	4.5

Dialog Settings
«Static Parameters» Define the static Thyristor parameters in the table. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Look the meaning up in the parameter table below.
«Dynamic Parameters» Define the dynamic Thyristor parameters in the table. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Look the meaning up in the parameter table below.



Component Parameters

Description [Unit]	Parameter Name	Data Type
Saturation Current [A], static parameter	IS	real
Forward Emission Coefficient, static parameter	NF	real
Reverse Emission Coefficient, static parameter	NR	real
Bulk Resistance Anode [Ω], static parameter	RA	real
Bulk Resistance Gate [Ω], static parameter	RG	real
Bulk Resistance Cathode [Ω], static parameter	RC	real
Amplification Factor, static parameter	K	real
Carrier lifetime [s], dynamic parameter	TT1	real
Carrier lifetime [s], dynamic parameter	TT2	real
Carrier lifetime [s], dynamic parameter	TT3	real
Junction Capacitance [F], dynamic parameter	CJO1	real
Junction Capacitance [F], dynamic parameter	CJO2	real
Junction Capacitance [F], dynamic parameter	CJO3	real

Component Nodes

Description	Node Name	Nature
Anode Pin	ANODE	electrical
Gate Pin	GATE	electrical
Cathode Pin	CATHODE	electrical

4.6 SPICE-Compatible Models

- Diodes
- Bipolar Junction Transistors
- Field-Effect Transistors
- Line Models

4.6.1 Introduction

SPICE-compatible models in SIMPLORER include all diode and transistor models provided in Berkeley SPICE 3f5, the lossy line model (LTRA), and the newest process-based MOSFET models (BSIM and EKV). A four-pin version of the BJT model is provided. The SPICE semiconductor capacitor and resistor models are not included. The models support both AC and DC analysis. SPICE-compatible models appear in the *Circuit* folder of the «Basics» tab.

Model Implementation

SPICE-compatible models are basic components at the device level. They run code from Berkeley SPICE 3f5, with additions for the BSIM3, BSIM4, and EKV model. The library interface to SIMPLORER mimics the SPICE circuit environment.

The device models in PSPICE, derived from an earlier version 2g6 of Berkeley SPICE, have evolved independently and they are comparable to those in SPICE 3f5, but they are not necessarily identical. A reference on these differences is *Semiconductor Device Modeling with SPICE*, 2nd ed., by Guiseppe Massobrio and Paulo Antognetti, McGraw-Hill, 1993. This book also describes the differences with HSPICE models.

Vendor-specific SPICE models are often implemented as a SPICE subcircuit composed of several basic components. The SPICE converter in SIMPLORER can handle the conversion of both basic component models, and subcircuit models.

A small number of model parameter names were changed to conform to SIMPLORER naming conventions. SPICE can sometimes accept two versions of a parameter name (for example VTO or VT0). SIMPLORER models accept only the preferred name (VTO). For transistor models, SPICE accepts a vector of initial voltages (for example IC with 3 elements), SIMPLORER models accept the alternative separate parameters for initial voltages (for example ICVDS, ICVGS, and ICVBS).

SPICE-compatible SIMPLORER models combine model and instance parameters in one component, whereas SPICE manages these parameter sets separately. SPICE also handles default parameters differently. See also “Default Parameter Handling” on page 151.

SPICE-compatible SIMPLORER models perform their own integration of charges and other state variables, always using the 2nd-order trapezoidal integration method. They also perform their own estimates of local truncation error, for use in limiting the time step. For these and other reasons, the SPICE-compatible models can behave differently than typical SIMPLORER models. See also “Initial Conditions” on page 152 and “Convergence Tolerances” on page 153.

Default Parameter Handling

For SPICE-compatible models, the default values were determined by creating a new instance of the device, and querying the library code for parameter values at the default temperature of 27 °C. A few of the default parameter values listed in the SPICE 3f5 documentation do not include the temperature adjustment. For example, the VTO parameter in MOS3, and the KP parameter in MOS2, both have default values of 20u listed in the SPICE 3f5 documentation. But SIMPLORER will report these default values as 20.7189u.

SPICE has the concept of a *not given* parameter, and by default no parameters are *given*. These parameters will still return a value when queried. For example, the diode breakdown voltage (BV) returns '0' when queried for a default value, but during simulation SPICE knows that BV was not specified and hence does not use it. The same effect can be simulated by specifying *infinity* for BV, as listed in the SPICE 3f5 documentation. The new SPICE-compatible models will return zero as default values for diode breakdown voltages and some of the BJT voltage parameters. In SPICE5 (model library in SIMPLORER 5), these default values were *infinity*, but the simulation behavior has not changed.

The *not given* and the *infinity default* approaches are not always equivalent. Sometimes there are side effects when parameters are not given - it may trigger more pre-processing in the SPICE model code. One example is the default U0=600 value from the SPICE 3f5 documentation for MOS1. This only applies if TOX is given, while KC and U0 are not given. Therefore SIMPLORER shows the default value as '0' instead of '600', because the simulation of a default MOS1 will behave as if TOX, KC, and U0 are all not given.

The Schematic passes all parameters to the SPICE-compatible model library, whether the user changed them or not. Internally, the library decides whether a parameter was given or not, by comparing to the default value. If you change a few parameters, there can be a discrepancy if the SPICE 3 pre-processing (side effects) are not correct. There is no way to specify a parameter equal to its default value. The library act as if it was not given and the pre-processing side effects may not be correct. This only applies to components dropped on the sheet and edited by the user. These problems do not occur with SML text in macros. If the macro specifies only a few parameters, the library should behave exactly as SPICE does. This would apply to any vendor-specific model converted by SIMPLORER.

Initial Conditions

SPICE has device-level initial conditions that are sometimes used to assist convergence, or to choose between multiple stable states. These options are available, but they behave differently in SIMPLORER than in SPICE:

1. Users can specify that a semiconductor device is *off* during the DC solution. This is mainly used to choose between multiple stable operating points. It is applicable to DC, AC, and TR analysis. The terminal voltages are still allowed to vary when the device is specified *off*. If SPICE cannot reach a converged DC solution, it will try a second time, ignoring the *off* specs. In SIMPLORER, the second attempt would not occur.
2. Users can specify initial semiconductor terminal voltages, but only for transient analysis. They are ignored for AC and DC. In SPICE, the DC solution before transient analysis is then skipped; any voltages left unspecified are assumed to be zero. The behavior for SIMPLORER is different – any model with a specified initial voltage will try to hold that voltage at the terminals. In other parts of the circuit, the initial voltages are not set to zero as in SPICE, instead, SIMPLORER solves for them as usual.

To employ option #1, set the OFF parameter for a device equal to '1'.

The device initial voltages (option #2) work, but it is easy to enter bad values and get bad solutions at time zero. The user is better off specifying initial voltages on external capacitors in the circuit. That will push the DC solution in the right direction. Initial voltages can also be specified on the LTRA model, at least when C is not zero. Please note that the preferred method in SPICE is to set initial node voltages, rather than device voltages. This method also works in SIMPLORER.

Convergence Tolerances

The library code adjusts the standard SPICE convergence options, based on the SIMPLORER simulation parameters. When running under SIMPLORER, SPICE-compatible models do not use these parameters for convergence testing, but they may use the parameters to reject a time step due to estimated local truncation error in the numerical integrations. The main SPICE parameter controlling this is RELTOL, which is directly comparable to the SIMPLORER LDF parameter.

SPICE Convergence Parameters

This summarizes SPICE 3f5 convergence parameters, specified on a .OPTIONS card.

- RELTOL – the relative error to be allowed in device voltage or current. Defaults to 0.1%, but users are often advised to make RELTOL=1%, either for faster simulations, or to achieve convergence in power electronic circuits. The SIMPLORER default for LDF is 1%.
- ABSTOL and VNTOL – minimum absolute error tolerances, sometimes called *best accuracy*, for current and voltage, respectively. These come into play when the voltages and currents approach zero, and RELTOL would produce a very low magnitude. Default values for SPICE are ABSTOL=1e-12 and VNTOL=1e-6. These must often be increased for power electronics, and users are advised to set values 8 or 9 orders of magnitude below the peak current or voltage expected. The SIMPLORER defaults for similar parameters VEmax and IEmax are both 1e-3, but these parameters do not mean the same as ABSTOL and VNTOL.
- CHGTOL – the *best accuracy* for charge on junction capacitances, independent of ABSTOL and VNTOL. Default value is 1e-14, often increased for power electronics.
- TRTOL – in SPICE, the local truncation error function estimates a time step, which is then multiplied by TRTOL for comparison to the present time step. Default value is 7, and it seems like a pure fudge factor. In Georgia Tech XSPICE, the default was changed to '1'. Users are sometimes advised to increase TRTOL, up to 40, for faster simulations by allowing larger time steps. There is no apparent counterpart in SIMPLORER.
- GMIN – the minimum conductance allowed in SPICE. Devices will not add matrix elements of lower magnitude than GMIN to the Jacobian. Defaults to 1e-12. There is no apparent counterpart in SIMPLORER.
- Various Iteration Limits – SIMPLORER takes care of the iteration limits for the SPICE models, so the SPICE iteration limits are not discussed here.

To assist convergence, SPICE users are often advised to loosen the tolerances, especially RELTOL. Increasing the iteration limits is also suggested.

Users are not often advised to tighten up the SPICE tolerances. An exception is with the voltage-controlled and current-controlled switches, where CHGTOL=1e-16 and TRTOL=1 has been suggested. The values should not be changed for this reason in SIMPLORER, because it does not use the SPICE switch models.

Besides their use for error estimates, nearly all of the SPICE device models refer to some of the convergence parameters internally when loading the Jacobian and the right-hand side. Sometimes this is used to bypass some calculations, and other times to adjust the matrix elements.

SPICE Tolerances in SIMPLORER

SPICE-compatible models set their convergence parameters from SIMPLORER:

- RELTOL is taken from LDF on the SIMPLORER TR parameters, scaling by 0.01 to convert LDF from percent. LDF is used even for the SIMPLORER DC analysis, because the same code executes both DC and Transient analysis in the SPICE models.
- ABSTOL is the default value scaled by SIMPLORER IEmax or Emax. For transient analysis, ABSTOL=1e-12 * IEmax / 1e-3, because 1e-3 is the SIMPLORER default value for IEmax. For DC analysis, including that which precedes AC analysis, ABSTOL = 1e-12 * EmaxDC / 1e-3. For AC analysis, ABSTOL = 1e-12 * EmaxAC / 1e-6, but it has no effect in the SPICE model execution.
- VNTOL is also scaled by SIMPLORER VEmax or Emax. For transient analysis, VNTOL = 1e-6 * VEmax / 1e-3. For DC analysis, VNTOL = 1e-6 / EmaxDC / 1e-3, and for AC analysis, VNTOL = 1e-6 / EmaxAC / 1e-6.
- CHGTOL is the default value scaled by the ABSTOL * VNTOL product. CHGTOL = 1e-14 * ABSTOL * VNTOL/1e-18.
- After getting ABSTOL, VNTOL, and CHGTOL, the minimum values are limited to 100 times less than their SPICE defaults. ABSTOLmin = 1e-14, VNTOLmin = 1e-8, and CHGTOLmin = 1e-16. There are no maximum values for ABSTOL, VNTOL, and CHGTOL.
- There are no minimum or maximum limits on RELTOL.
- GMIN is fixed at 1e-12, and TRTOL is fixed at 7.

Recommendations

The main parameter for users to adjust would be LDF, just as the main SPICE parameter to adjust would be RELTOL. The suggested starting value is 1%, and later decreasing to 0.1% for more accuracy. The default behavior of the SPICE-compatible models will be RELTOL=0.01 instead of RELTOL=0.001, but all other default convergence parameters are the same as in SPICE 3f5.

The DC analysis of a SPICE-compatible model executes the same code as for a transient analysis, and it will use RELTOL (LDF) from the SIMPLORER transient parameters. This value should be reviewed when running a DC analysis.

In most cases, the SIMPLORER default values of 1e-3 for IEmax and VEmax will work with the SPICE-compatible models. However, it may be necessary to reduce them for acceptable results in some cases.

SPICE 3f5 users are sometimes advised to allow up to 500 iterations for DC and transient analysis. If convergence becomes a problem, these higher limits can be set directly in the SIMPLORER simulation parameter `ITERATMAX`.

Temperature Effects on SPICE-Compatible Models

SPICE semiconductor device parameters can be adjusted for temperature. If you define the component temperature different from the default value of 27 °C, that temperature is used. Otherwise the model ambient temperature, which is defined as simulation parameter, is used.

If you change the model ambient temperature in SIMULATION>PARAMETERS **«General»**, but you want a component to stay at the default value of 27, then you must change that component temperature to a slightly different value, such as 27.01. This is necessary of how the library must handle default values.

4.6.2 Diode Models

The diode components are implemented as C models. The models use the *MaxwellSpice.dll* file. The simulation parameter HMIN (minimum time step) should be 0.1ns for the first simulation run. Then you can adjust the value for optimal simulation performance. For simulations of power electronics circuits, HMIN can be up to 100ns.

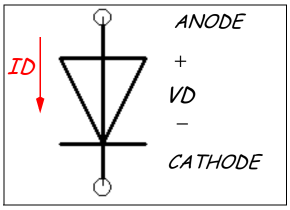
Diode

>>Basics>Circuit>Spice-Compatible Models/Diodes

The Spice-Compatible Diode Model simulates the electrical and thermal behavior of a diode using SPICE3 parameters and equations. Electrical characteristics and the temperature dependence of diode behavior are calculated according to user-defined of device parameters.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

4.6.3 Bipolar Junction Transistors

The BJT components are implemented as C models. The models use the *MaxwellSpice.dll* file.

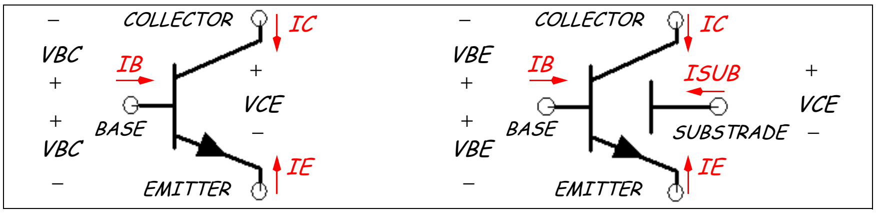
The simulation parameter Hmin (minimum time step) should be 0.1ns for the first simulation run. Then you can adjust the value for optimal simulation performance. For simulations of power electronics circuits, Hmin can be up to 100ns.

BJT Models

>>Basics>Circuit>Spice-Compatible Models>Bipolar Junction Transistors
The Spice-compatible BJT NPN-type, BJT PNP-type, BJT4 NPN-type, and BJT4 PNP-type models simulate the electrical and thermal behavior of a bipolar junction transistor using SPICE3 parameters and equations. Electrical characteristics and the temperature dependence of transistor behavior are calculated according to user-defined device parameters.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

4.6.4 Field-Effect Transistors

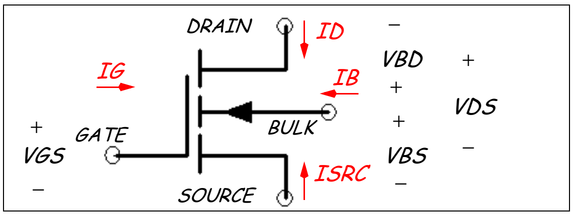
The field-effect transistor components are implemented as C models. The models use the *MaxwellSpice.dll* file. The simulation parameter HMIN (minimum time step) should be 0.1ns for the first simulation run. Then you can adjust the value for optimal simulation performance. For simulations of power electronics circuits, HMIN can be up to 100ns.

MOS1 Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors
The Spice-Compatible MOS1 N-channel and MOS1 P-channel models simulate the electrical and thermal behavior of MOSFETs using the parameters and equations of the SPICE3 Level 1 MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-defined device parameters.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
⭳⭱	«Value» Click the «Value» field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

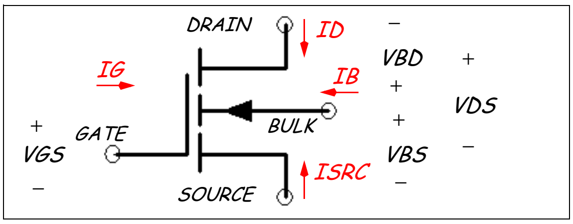
MOS2 Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors

The Spice-Compatible MOS2 N-channel and MOS2 P-channel models simulate the electrical and thermal behavior of MOSFETs using the parameters and equations of the SPICE3 Level 2 MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-defined device parameters.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
⭳⭱	«Value» Click the «Value» field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help).

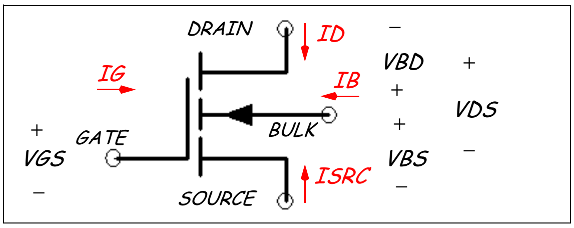
«Name»«Default»«Unit»«Description»
Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

MOS3 Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors
The Spice-Compatible MOS3 N-channel and MOS3 P-channel models simulate the electrical and thermal behavior of MOSFETs using the parameters and equations of the SPICE3 Level 3 MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-defined device parameters.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings

«Value»
Click the «**Value**» field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation.

Look the parameter meaning up in the info lines of the parameter table (online help).

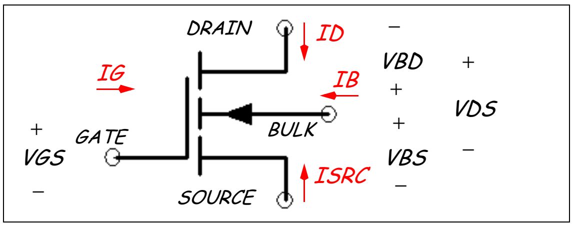
«Name»«Default»«Unit»«Description»
Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

MOS6 Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors
The Spice-Compatible MOS6 N-channel and MOS6 P-channel models simulate the electrical and thermal behavior of MOSFETs using the parameters and equations of the SPICE3 Level 6 MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-defined device parameters.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

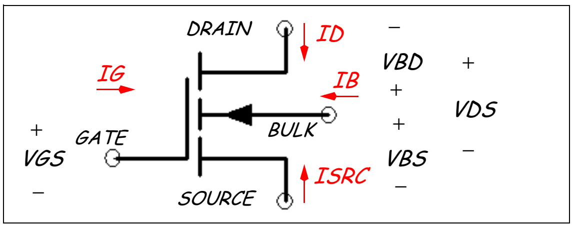
BSIM1 Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors

The Spice-Compatible BSIM1 N-channel and BSIM1 P-channel models simulate the electrical and thermal behavior of small-geometry MOSFETs using the parameters and equations of the SPICE3 BSIM1 MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-defined device parameters that are obtained from MOSFET process characterization.This model is also referred to as MOS level 4.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help).

	«Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

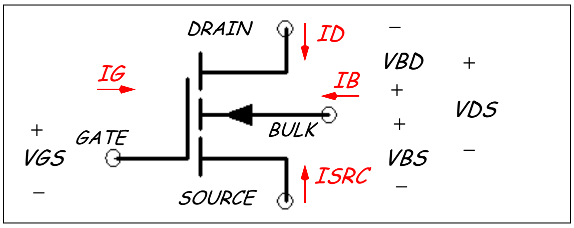
BSIM2 Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors

The Spice-Compatible BSIM2 N-channel and BSIM2 P-channel models simulate the electrical and thermal behavior of small-geometry MOSFETs using the parameters and equations of the BSIM2 MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-defined device parameters that are obtained from MOSFET process characterization. This model is also referred to as MOS level 5.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). **«Name»«Default»«Unit»«Description»** Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

BSIM3 Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors

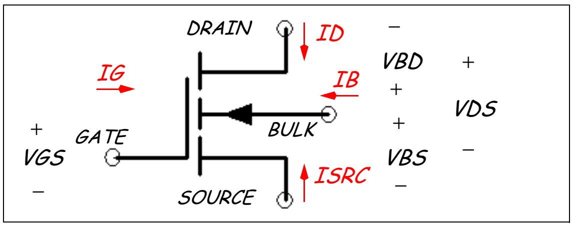
The Spice-Compatible BSIM3 N-channel and BSIM3 P-channel models simulate the electrical and thermal behavior of small-geometry MOSFETs using the parameters and equations of the BSIM3 MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-input device parameters that are obtained from MOSFET process characterization.

See also information on the Web site of UC Berkeley Device Group at

http://www-device.eecs.berkeley.edu/~bsim3/

BSIM3 and BSIM4 are developed by the Device Research Group of the Department of Electrical Engineering and Computer Science, University of California, Berkeley and copyrighted by the University of California.

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

BSIM4 Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors

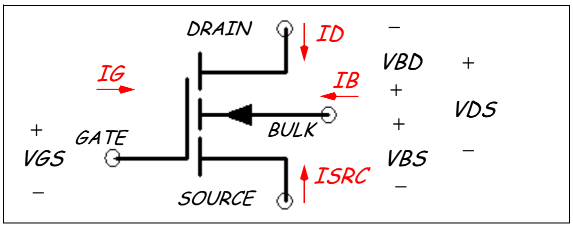
The Spice-Compatible BSIM4 N-channel and BSIM4 P-channel models simulate the electrical and thermal behavior of small-geometry MOSFETs using the parameters and equations of the BSIM4 MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-defined device parameters that are obtained from MOSFET process characterization.

See also information on the Web site of UC Berkeley Device Group at

http://www-device.eecs.berkeley.edu/~bsim3/

BSIM3 and BSIM4 are developed by the Device Research Group of the Department of Electrical Engineering and Computer Science, University of California, Berkeley and copyrighted by the University of California.

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

EKV Model

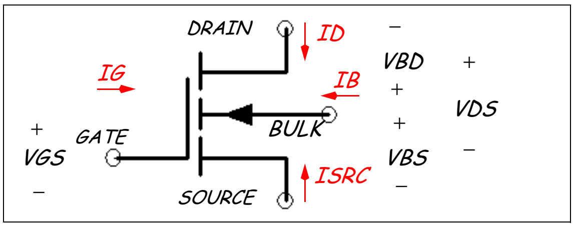
>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors

The Spice-Compatible EKV N-channel and EKV P-channel models simulate the electrical and thermal behavior of submicron MOSFETs used in low-voltage, low-current analog and mixed signal circuits. The Spice-Compatible EKV models use the parameters and equations of the EKV MOSFET model. Electrical characteristics and the temperature dependence of MOSFET behavior are calculated according to user-defined device parameters that are based on fundamental physical properties of the MOS structure.

See also information on the Web site of Electronics Laboratory of EPFL (Lausanne) at

http://legwww.epfl.ch/ekv/

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

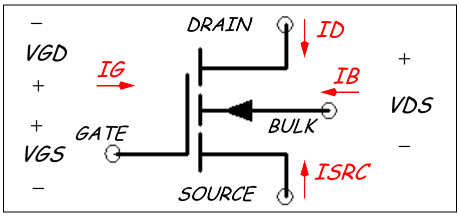
MESFET Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors

The Spice-compatible MESFET N-channel and MESFET P-channel models simulate the electrical and thermal behavior of a JFET using SPICE3 parameters and equations. Electrical characteristics and the temperature dependence of MESFET behavior are calculated according to user-defined device parameters.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

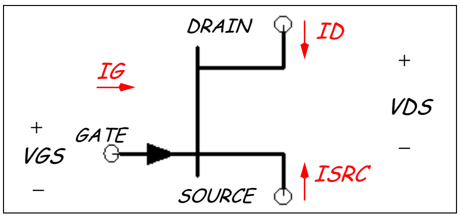
JFET Model

>>Basics>Circuit>Spice-Compatible Models>Field-Effect Transistors

The Spice-compatible JFET N-channel and JFET P-channel models simulate the electrical and thermal behavior of a JFET using SPICE3 parameters and equations. Electrical characteristics and the temperature dependence of JFET behavior are calculated according to user-defined device parameters.

See also Massobrio, Giueseppe, and Paolo Antognetti: *Semiconductor Device Modeling with SPICE*, New York: McGraw Hill, 1993

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help).
	«Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

4.6.5 Line Models

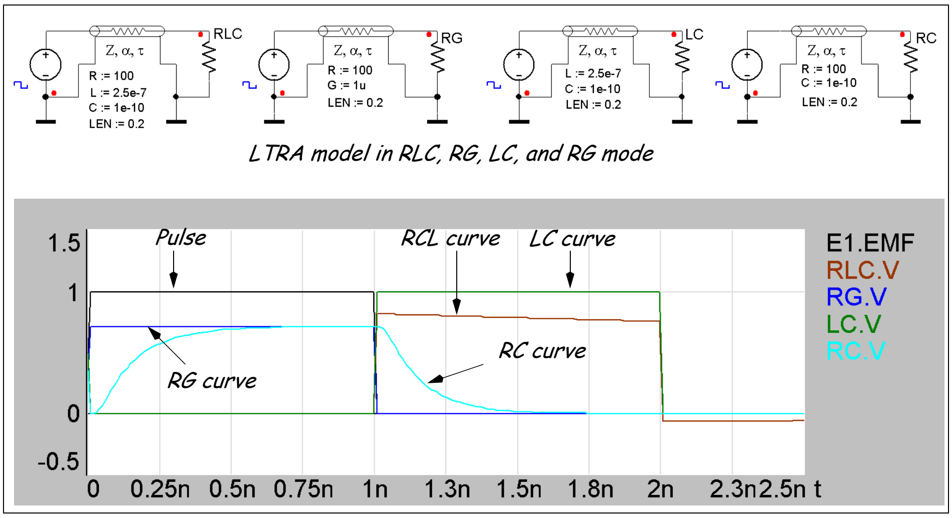
The line components are implemented as C models. The models use the *MaxwellSpice.dll* file.

The simulation parameter `HMIN` (minimum time step) should be 0.1ns for the first simulation run. Then you can adjust the value for optimal simulation performance. For simulations of power electronics circuits, `HMIN` can be up to 100ns. Large values of `HMIN` can be used for long lines.

Lossy Transmission Line

>>Basics>Circuit>Spice-Compatible Models>Line Models
The Spice-compatible lossy transmission line model simulates the electrical behavior of a lossy transmission line using SPICE3 LTRA parameters and equations. Line electrical characteristics are calculated according to user-defined line parameters.

The example shows all four supported modes of the LTRA model. The line is 0.2m long, and it has a characteristic impedance given by sqrt (L/C) = 50 Ω. The wave velocity is given by 1/sqrt (LC)=2e8 m/s, and the line's travel time is 1ns. With a series resistance of 100 Ω/m, the total series resistance is 20 Ω, or 40% of the characteristic impedance, so the line is very lossy.



The input voltage is a 1V pulse, 1ns wide. For the case of R=0, the input appears at the far end, delayed by 1ns (LC curve). There is no attenuation because the line R is zero. There is no reflection because the terminating resistance is equal to the characteristic impedance.

With high series resistance, the input is delayed 1ns as before, but distortion and attenuation occur (RLC curve). With only series resistance and a very low shunt conductance, there is no delay or distortion, but the output is attenuated (RG curve). Finally, with L=0 there is a delay from the RC time constant, but the delay is not 1-ns (RC curve).

When the input pulse is very narrow compared to the line's travel time, and the time step increases during simulation, it is possible for the pulse to become *lost*. It never appears at the far end. The *breakpoint* mechanism in the LTRA model does not fully address this problem, even when running in SPICE itself. A very small resistance value (for example R=1u) usually solves this problem.

See also Roychowdhury, J. S., and D. O. Pederson: *Efficient Transient Simulation of Lossy Interconnect*, San Francisco, CA: ACM/IEEE Design Automation Conference, June 17-21, 1991

Line Model Applications

SPICE 3f5 implements three different line models:

- LTRA - the lossy transmission line model, implemented with convolution of the impulse response.
- T - the lossless transmission line, implementing an idealized traveling wave behavior.
- URC - the uniform RC line, sometimes useful in low-frequency applications.

For the lossless case, LTRA is typically more accurate and efficient than the older T model. The URC model automatically creates an internal lumped RC network for SPICE, which is rather inefficient, and again, less accurate than the truly distributed-parameter LTRA model. Therefore, only the LTRA model has been provided for SIMPLORER. All instances of the T and URC model should be converted to use the LTRA model.

Converting Lossless T Components to LTRA

The basic T component parameters are the characteristic impedance, and the line travel or delay time. To derive LTRA parameters, it's necessary to assume a wave velocity. For example, 3e8 m/s (speed of light) will be assumed here. A different velocity may be assumed, and the derived parameters would be different but produce the same results in circuit simulation.

The characteristic impedance is directly available as Z0. The travel time may be input directly as TD, or indirectly as a frequency and normalized line length. If necessary, obtain the travel time as:

- TD = NRMLEN / FREQ

If not specified, NRMLEN = 0.25 for a quarter wavelength.

Next, obtain the length and the other LTRA parameters as:

- LEN = 3e8 * TD
- L = Z0 / 3e8
- C = 1 / (Z0 * 3e8)
- R = G = 0

For example, given Z0 = 50 and TD = 1n, we would obtain LEN = 0.3, L = 1.667e-7, and C = 6.667e-11. Assuming a velocity of 2e8 instead of 3e8, we would obtain LEN = 0.2, L = 2.5e-7, and C = 1e-10 as in the example of Figure 2. Both parameter sets would give the same results in SIMPLORER.

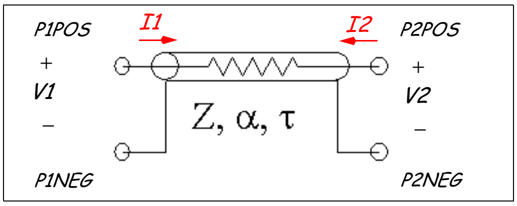
Converting Uniform RC Components to LTRA

The LTRA parameters should be:

- LEN = LEN from the URC instance
- R = RPERL from the URC model
- C = CPERL from the URC model
- L = G = 0

Other parameters of the URC model should be ignored. The diode mode of URC, which employs ISPERL and RSPERL parameters, is not supported for SIMPLORER.

Current and Voltage Reference Arrow System



Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. The parameters can be set only at the beginning of a simulation. Look the parameter meaning up in the info lines of the parameter table (online help). «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

4.7 Electrical Machines

- Induction Machine (ASM)
- Synchronous Machine Electrical Excitation without Damper (SYME)
- Synchronous Machine Electrical Excitation with Damper (SYMED)
- Synchronous Machine Permanent Excitation without Damper (SYMP)
- Synchronous Machine Permanent Excitation with Damper (SYMPD)
- DC Machine Electrical Excitation (GSME)
- DC Machine Electrical Nonlinear Excitation (GSMENL)
- DC Machine Permanent Excitation (GSMP)

Induction Machine with Squirrel Cage Rotor

>>Basics>Circuit> Electrical Machines

The model represents an induction machine with squirrel cage rotor and star-connected stator wingdings as a lumped circuit component. The circuit nodes A–B–C are the terminals of star-connected stator windings. The component cannot be used with AC and DC simulation.

If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the induction machine are of special interest, voltmeters or ammeters can be connected to the induction machine.

Model Limits of Induction Machine Model

- 3-phase symmetrical induction machine with squirrel cage rotor and star-connected stator wingdings without neutral node (no zero phase-sequence system).
- Linear and iron-loss free magnetic circuit.
- No consideration of skin effects in the wingdings (restricted simulation accuracy at e.g. start-up processes; typical case: current-displacement motor connected to the mains)
- Exclusive consideration of fundamental flux linkage between stator and rotor wingdings
- Rotor position-independent leakage inductances.
- Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally.

Equation System

The equation system is implemented in a stator-fixed coordinate system (α-β coordinates). Index 1 represents the stator quantities, index 2 the rotor quantities. The phase quantities of the real three-phase induction machine are indicated with a, b, c.

Voltage equations

$$v_{1\alpha}(t) = i_{1\alpha}(t) \cdot R_1 + \frac{d\Psi_{1\alpha}(t)}{dt} \qquad 0 = i_{2\alpha}(t) \cdot R_2 + \frac{d\Psi_{2\alpha}(t)}{dt} + p \cdot \omega(t) \cdot \Psi_{2\beta}(t)$$

$$v_{1\beta}(t) = i_{1\beta}(t) \cdot R_1 + \frac{d\Psi_{1\beta}(t)}{dt} \qquad 0 = i_{2\beta}(t) \cdot R_2 + \frac{d\Psi_{2\beta}(t)}{dt} - p \cdot \omega(t) \cdot \Psi_{2\alpha}(t)$$

Flux-linkage equations

$$\Psi_{1\alpha}(t) = i_{1\alpha}(t) \cdot L_1 + i_{2\alpha}(t) \cdot L_m$$
$$\Psi_{1\beta}(t) = i_{1\beta}(t) \cdot L_1 + i_{2\beta}(t) \cdot L_m$$
$$\Psi_{2\alpha}(t) = i_{1\alpha}(t) \cdot L_m + i_{2\alpha}(t) \cdot L_2$$
$$\Psi_{2\beta}(t) = i_{1\beta}(t) \cdot L_m + i_{2\beta}(t) \cdot L_2$$

4

Torque equation (electromagnetic developed "internal" torque)

$$m_i(t) = \frac{3}{2}p \cdot (\Psi_{1\alpha}(t) \cdot i_{1\beta}(t) - \Psi_{1\beta}(t) \cdot i_{1\alpha}(t))$$

Motion equation

$$\frac{d\omega(t)}{dt} = \frac{1}{J}(m_i(t) - m_w(t))$$

Electrical coupling of phase quantities with the orthogonal quantities

$$v_{1\beta}(t) = \frac{\sqrt{3}}{2}v_{1b}(t) - \frac{\sqrt{3}}{3}v_{1c}(t)$$

$$v_{1\alpha}(t) = \frac{2}{3}v_{1a}(t) - \frac{1}{3}v_{1b}(t) - \frac{1}{3}v_{1c}(t)$$

$$i_{1a}(t) = i_{1\alpha}(t)$$

$$i_{1b}(t) = -\frac{1}{2}i_{1\alpha}(t) + \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

$$i_{1c}(t) = -\frac{1}{2}i_{1\alpha}(t) - \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

Dialog Settings	
↓↑	«Parameters» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
↓↑	«Initial Values» Click the **«Value»** field of the corresponding parameter to define the initial values. Common parameter types. Enter a numerical value, a variable, or an expression. The values are set only once at simulation start. Look the meaning up in the parameter table below.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension. The dimension is valid for all angle input quantities in the machine model.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Load Torque [Nm]	LOAD	real
Stator Resistance [Ω]	R1	real
Rotor Resistance [Ω] related to stator side	R2	real
Stator Leakage Inductance [H] leakage inductance of stator side	LS1	real
Rotor Leakage Inductance [H] leakage inductance of rotor side	LS2	real
Main Inductance [H] coupling inductance between stator and rotor side	LM	real
Number of Pole Pairs [/]	P	real
Moment of Inertia [kg*m^2]	J	real
Initial Current Stator Phase a [A]	I1A0	real
Initial Current Stator Phase b [A]	I1B0	real
Initial Current Stator Phase c [A]	I1C0	real
Initial Current Rotor Phase a [A] with ratio of induction machine convert to stator side	I2A0	real

Initial Current Rotor Phase b [A] with ratio of induction machine convert to stator side	I2B0	real
Initial Current Rotor Phase c [A] with ratio of induction machine convert to stator side	I2C0	real
Initial Rotor Speed [rpm]	N0	real
Initial Rotor Position [deg]	PHI0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Rotor Speed [rpm]	N	real
Inner Torque [Nm]	MI	real
Rotor rational Speed [1/s]	OMEGA	real
Angular Rotor Acceleration [1/s^2]	ALPHA	real
Rotor Angle in Radians [rad]	PHI	real
Rotor Angle in Degrees [deg]	PHIDEG	real
Rotor Angle Mechanical in Radians [rad]	PHIM	real
Rotor Angle Mechanical in Degrees [deg]	PHIMDEG	real
Current Stator Phase a [A]	I1A	real
Current Stator Phase b [A]	I1B	real
Current Stator Phase c [A]	I1C	real
Current Rotor Phase a [A]	I2A	real
Current Rotor Phase b [A]	I2B	real
Current Rotor Phase c [A]	I2C	real
Linked Stator Flux Alpha Component [Vs]	PSI1A	real
Linked Stator Flux Beta Component [Vs]	PSI1B	real
Linked Rotor Flux Alpha Component [Vs]	PSI2A	real
Linked Rotor Flux Beta Component [Vs]	PSI2B	real
1st derivative Stator Phase Current a [A/s]	dI1A	real
1st derivative Stator Phase Current b [A/s]	dI1B	real
1st derivative Stator phase Current c [A/s]	dI1C	real
1st derivative Rotor Phase Current a [A/s]	dI2A	real
1st derivative Rotor Phase Current b [A/s]	dI2B	real
1st derivative Rotor Phase Current c [A/s]	dI2C	real
1st derivative of linked Stator Flux alpha component [V]	dPSI1A	real
1st derivative of linked Stator Flux beta component [V]	dPSI1B	real
1st derivative of linked Rotor Flux alpha component [V]	dPSI2A	real
1st derivative of linked Rotor Flux beta component [V]	dPSI2B	real

Component Nodes

Description	Node Name	Nature
Stator Node A	A	electrical
Stator Node B	B	electrical
Stator Node C	C	electrical

Synchronous Machine Electrical Excitation without Damper

>>Basics>Circuit>Electrical Machines

The model represents a linear DC field synchronous machine without damper (internal-field) as a lumped circuit component. Depending on the parameter set, the machine can be operated either as a salient or non-salient pole rotor. The circuit nodes A – B – C are the terminals of star-connected stator winding; the circuit nodes E1 – E2 are the terminals of excitation winding. The component cannot be used with AC and DC simulation.

If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the synchronous machine are of special interest, voltmeters or ammeters can be connected to the synchronous machine.

Model Limits of Synchronous Machine Models

- 3-phase symmetrical synchronous machine with internal-field system and star-connected stator winding without neutral node (no zero phase-sequence system).
- Linear and iron-loss free magnetic circuit.
- No consideration of skin effects in the windings.
- Exclusive consideration of fundamental flux linkage between stator, damper and excitation windings.
- Rotor position-independent leakage inductances.
- Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally.

Equation System

The equation system is implemented in a rotor-fixed (rotor-fixed and also rotor flux-fixed) coordinate system (d-q coordinates). Index *1* represents the stator quantities; the index *e* the quantities of rotor excitation wingdings. The phase quantities of the real three-phase synchronous machine are indicated with *a*, *b*, *c*.

Voltage equations

$$v_{1d}(t) = i_{1d}(t) \cdot R_1 + \frac{d\psi_{1d}(t)}{dt} - p \cdot \omega(t) \cdot \psi_{1q}(t) \qquad v_{ed}(t) = i_{ed}(t) \cdot R_e + \frac{d\psi_{ed}(t)}{dt}$$

$$v_{1q}(t) = i_{1q}(t) \cdot R_1 + \frac{d\psi_{1q}(t)}{dt} + p \cdot \omega(t) \cdot \psi_{1d}(t) \qquad v_{eq}(t) = 0$$

Flux-linkage equations

$$\psi_{1d}(t) = i_{1d}(t) \cdot L_{1d} + i_{ed}(t) \cdot L_{m1ed}$$
$$\psi_{1q}(t) = i_{1q}(t) \cdot L_{1q}$$
$$\psi_{ed}(t) = i_{1d}(t) \cdot L_{m1ed} + i_{ed}(t) \cdot L_{ed}$$
$$\psi_{eq}(t) = 0$$

Torque equation (electromagnetic developed "internal" torque)

$$m_i(t) = \frac{3}{2} p \cdot (\psi_{1d}(t) \cdot i_{1q}(t) - \psi_{1q}(t) \cdot i_{1d}(t))$$

Motion equation

$$\frac{d\omega(t)}{dt} = \frac{1}{J}(m_i(t) - m_w(t))$$

Electrical coupling of excitation quantities with the orthogonal quantities

$$v_{ed}(t) = v_e(t) \qquad i_e(t) = i_{ed}(t)$$

Electrical coupling of phase quantities with the orthogonal quantities

$$\gamma(t) = p \cdot \int \omega(t)dt$$

$$v_{1\alpha}(t) = \frac{2}{3}v_{1a}(t) - \frac{1}{3}v_{1b}(t) - \frac{1}{3}v_{1c}(t)$$

$$v_{1\beta}(t) = \frac{\sqrt{3}}{3}v_{1b}(t) - \frac{\sqrt{3}}{3}v_{1c}(t)$$

$$v_{1d}(t) = v_{1\alpha}(t) \cdot \cos(\gamma(t)) + v_{1\beta}(t) \cdot \sin(\gamma(t))$$

$$v_{1q}(t) = v_{1\alpha}(t) \cdot \sin(\gamma(t)) - v_{1\beta} \cdot \cos\gamma(t)$$

$$i_{1\alpha}(t) = i_{1d}(t) \cdot \cos(\gamma(t)) - i_{1q}(t) \cdot \sin(\gamma(t))$$

$$i_{1\beta}(t) = i_{1d}(t) \cdot \sin(\gamma(t)) + i_{1q}(t) \cdot \cos(\gamma(t))$$

$$i_{1a}(t) = i_{1\alpha}(t)$$

$$i_{1b}(t) = -\frac{1}{2}i_{1\alpha}(t) + \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

$$i_{1c}(t) = -\frac{1}{2}i_{1\alpha}(t) - \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

Dialog Settings	
	«Parameters» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
	«Initial Values» Click the **«Value»** field of the corresponding parameter to define the initial values. Common parameter types. Enter a numerical value, a variable, or an expression. The values are set only once at simulation start. Look the meaning up in the parameter table below.
	«Angular Dimension» In the list, select radians or degrees for the angular dimension. The dimension is valid for all angle input quantities in the machine model.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Load Torque [Nm]	LOAD	real
Stator Resistance [Ω]	R1	real
Stator Inductance d-Axis [H]	L1D	real
Stator Inductance q-Axis [H]	L1Q	real
Mutual Inductance stator-exciter d-Axis [H]	LM1ED	real
Excitation Resistance [Ω]	RE	real
Excitation Inductance d-Axis [H]	LE	real
Number of Pole Pairs [/]	P	real
Moment of Inertia [kg*m²]	J	real
Initial Current Stator Phase a [A]	I1A0	real
Initial Current Stator Phase b [A]	I1B0	real
Initial Current Stator Phase c [A]	I1C0	real
Initial Excitation Current [A]	IE0	real

Initial Rotor Speed [rpm]	N0	real
Initial Rotor Position [deg]	PHI0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Rotor Speed [rpm]	N	real
Inner Torque [Nm]	MI	real
Rotor rational Speed [1/s]	OMEGA	real
Angular Rotor Acceleration [1/s^2]	ALPHA	real
Rotor Angle in Radians [rad]	PHI	real
Rotor Angle in Degrees [deg]	PHIDEG	real
Rotor Angle Mechanical in Radians [rad]	PHIM	real
Rotor Angle Mechanical in Degrees [deg]	PHIMDEG	real
Current Stator Phase a [A]	I1A	real
Current Stator Phase b [A]	I1B	real
Current Stator Phase c [A]	I1C	real
Excitation Current d-Axis [A]	IE	real
Real Part of Stator Current [A]	I1D	real
Imaginary Part of Stator Current [A]	I1Q	real
Real Part of Stator Voltage [V]	V1D	real
Imaginary Part of Stator Voltage [V]	V1Q	real
Real Part of Stator Flux Linkage [Vs]	PSI1D	real
Imaginary Part of Stator Flux Linkage [Vs]	PSI1Q	real
Excitation Flux Linkage d-Axis [Vs]	PSIE	real

Component Nodes

Description	Node Name	Nature
Stator Node A	A	electrical
Stator Node B	B	electrical
Stator Node C	C	electrical
Excitation Circuit Node E1	E1	electrical
Excitation Circuit Node E2	E2	electrical

Synchronous Machine Electrical Excitation with Damper

>>Basics>Circuit> Electrical Machines

The model represents a linear DC field synchronous machine with damper as a lumped circuit component. The damper circuit is not accessible from outside the machine; the corresponding phase windings are connected at a virtual neutral node. Depending on the parameter set, the machine can be operated either as a salient or non-salient pole rotor. The circuit nodes A – B – C are the terminals of star-connected stator winding; the circuit nodes E1 – E2 are the terminal of excitation winding. The component cannot be used with AC and DC simulation.

If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the synchronous machine are of special interest, voltmeters or ammeters can be connected to the synchronous machine.

In the case of identical parameters for the inductances L1D and L1Q, a synchronous machine with permanent magnet and non-salient pole rotor is modeled. To model a permanent magnet salient-pole machine, the parameters must be different for L1D and L1Q. See also “Model Limits of Synchronous Machine Models” on page 170.

Equation System

The equation system is implemented in a rotor-fixed (rotor-fixed and also rotor flux-fixed) coordinate system (d-q coordinates). Index *1* represents the stator quantities; the index *2* the equivalent quantities of the damper circuit belonged to the rotor. The phase quantities of the real three-phase synchronous machine are indicated with *a*, *b*, *c*.

Voltage equations

$$v_{1d}(t) = i_{1d}(t)\cdot R_1 + \frac{d\psi_{1d}(t)}{dt} - p\cdot\omega(t)\cdot\psi_{1q}(t) \qquad 0 = i_{e2}(t)\cdot R_2 + \frac{d\psi_{2d}(t)}{dt}$$

$$v_{1q}(t) = i_{1q}(t)\cdot R_1 + \frac{d\psi_{1q}(t)}{dt} + p\cdot\omega(t)\cdot\psi_{1d}(t) \qquad 0 = i_{2q}(t)\cdot R_2 + \frac{d\psi_{2q}(t)}{dt}$$

$$v_{ed}(t) = i_{ed}(t)\cdot R_e + \frac{d\psi_{ed}(t)}{dt} \qquad v_{eq}(t) = 0$$

Flux-linkage equations

$$\psi_{1d}(t) = i_{1d}(t)\cdot L_{1d} + i_{2d}(t)\cdot L_{m12d} + i_{ed}(t)\cdot L_{m1ed}$$
$$\psi_{1q}(t) = i_{1q}(t)\cdot L_{1q} + i_{2d}(t)\cdot L_{m12q}$$
$$\psi_{2d}(t) = i_{1d}(t)\cdot L_{m12d} + i_{2d}(t)\cdot L_{2d} + i_{ed}(t)\cdot L_{m2ed}$$
$$\psi_{2q}(t) = i_{1q}(t)\cdot L_{m12q} + i_{2q}(t)\cdot L_{2q}$$
$$\psi_{ed}(t) = i_{1d}(t)\cdot L_{m1ed} + i_{2d}(t)\cdot L_{m2ed} + i_{ed}(t)\cdot L_{ed}$$
$$\psi_{eq}(t) = 0$$

Torque equation (electromagnetic developed “internal” torque)

$$m_i(t) = \frac{3}{2}p\cdot(\psi_{1d}(t)\cdot i_{1q}(t) - \psi_{1q}(t)\cdot i_{1d}(t))$$

Motion equation

$$\frac{d\omega(t)}{dt} = \frac{1}{J}(m_i(t) - m_w(t))$$

Electrical coupling of excitation quantities with the orthogonal quantities

$$v_{ed}(t) = v_e(t) \qquad i_e(t) = i_{ed}(t)$$

Electrical coupling of phase quantities with the orthogonal quantities

$$\gamma(t) = p \cdot \int \omega(t)dt$$

$$v_{1\alpha}(t) = \frac{2}{3}v_{1a}(t) - \frac{1}{3}v_{1b}(t) - \frac{1}{3}v_{1c}(t)$$

$$v_{1\beta}(t) = \frac{\sqrt{3}}{3}v_{1b}(t) - \frac{\sqrt{3}}{3}v_{1c}(t)$$

$$v_{1d}(t) = v_{1\alpha}(t) \cdot \cos(\gamma(t)) + v_{1\beta}(t) \cdot \sin(\gamma(t))$$

$$v_{1q}(t) = v_{1\alpha}(t) \cdot \sin(\gamma(t)) - v_{1\beta} \cdot \cos\gamma(t)$$

$$i_{1\alpha}(t) = i_{1d}(t) \cdot \cos(\gamma(t)) - i_{1q}(t) \cdot \sin(\gamma(t))$$

$$i_{1\beta}(t) = i_{1d}(t) \cdot \sin(\gamma(t)) + i_{1q}(t) \cdot \cos(\gamma(t))$$

$$i_{1a}(t) = i_{1\alpha}(t)$$

$$i_{1b}(t) = -\frac{1}{2}i_{1\alpha}(t) + \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

$$i_{1c}(t) = -\frac{1}{2}i_{1\alpha}(t) - \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

Dialog Settings	
	«Parameters» Click the «Value» field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
	«Initial Values» Click the «Value» field of the corresponding parameter to define the initial values. Common parameter types. Enter a numerical value, a variable, or an expression. The values are set only once at simulation start. Look the meaning up in the parameter table below.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension. The dimension is valid for all angle input quantities in the machine model.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Load Torque [Nm]	LOAD	real
Stator Resistance [Ω]	R1	real
Damper Resistance [Ω]	R2	real
Stator Inductance d-Axis [H] Leakage inductance of stator side	L1D	real
Stator Inductance q-Axis [H] Leakage inductance of stator side	L1Q	real
Damper Inductance d-Axis [H] Leakage inductance of damper circuit side	L2D	real
Damper Inductance q-Axis [H] Leakage inductance of damper circuit side	L2Q	real
Mutual Inductance stator-damper d-Axis [H]	LM1D	real
Mutual Inductance stator-damper q-Axis [H]	LM12Q	real
Excitation Resistance [Ω]	RE	real
Excitation Inductance d-Axis [H]	LE	real
Mutual Inductance stator-exciter d-Axis [H]	LM1ED	real
Mutual Inductance damper-exciter q-Axis [H]	LM2EQ	real
Number of Pole Pairs [/]	P	real
Moment of Inertia [kg*m^2]	J	real

Initial Current Stator Phase a [A]	I1A0	real
Initial Current Stator Phase b [A]	I1B0	real
Initial Current Stator Phase c [A]	I1C0	real
Initial Current Damper Phase a [A]	I2A0	real
Initial Current Damper Phase b [A]	I2B0	real
Initial Current Damper Phase c [A]	I2C0	real
Initial Excitation Current [A]	IE0	real
Initial Rotor Speed [rpm]	N0	real
Initial Rotor Position [deg]	PHI0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Rotor Speed [rpm]	N	real
Inner Torque [Nm]	MI	real
Rotor rational Speed [1/s]	OMEGA	real
Angular Rotor Acceleration [$1/s^2$]	ALPHA	real
Rotor Angle in Radians [rad]	PHI	real
Rotor Angle in Degrees [deg]	PHIDEG	real
Rotor Angle Mechanical in Radians [rad]	PHIM	real
Rotor Angle Mechanical in Degrees [deg]	PHIMDEG	real
Current Stator Phase a [A]	I1A	real
Current Stator Phase b [A]	I1B	real
Current Stator Phase c [A]	I1C	real
Current Damper Phase a [A]	I2A	real
Current Damper Phase b [A]	I2B	real
Current Damper Phase c [A]	I2C	real
Excitation Current [A]	IE	real
Real Part of Stator Current [A]	I1D	real
Imaginary Part of Stator Current [A]	I1Q	real
Real Part of Damper Circuit Current [A]	I2D	real
Imaginary Part of Damper Circuit Current [A]	I2Q	real
Real Part of Stator Voltage [V]	V1D	real
Imaginary Part of Stator Voltage [V]	V1Q	real
Real Part of Stator Flux Linkage [Vs]	PSI1D	real
Imaginary Part of Stator Flux Linkage [Vs]	PSI1Q	real
Real Part of Damper Circuit Flux Linkage [Vs]	PSI2D	real
Imaginary Part of Damper Circuit Flux Linkage [Vs]	PSI2Q	real
Excitation Flux [Vs]	PSIE	real

Component Nodes

Description	Node Name	Nature
Stator Node A	A	electrical
Stator Node B	B	electrical
Stator Node C	C	electrical
Excitation Circuit Node E1	E1	electrical
Excitation Circuit Node E2	E2	electrical

Synchronous Machine Permanent Excitation without Damper

>>Basics>Circuit>Electrical Machines

The model represents a synchronous machine with permanent magnet excitation and without damper as a lumped circuit component. Depending on the parameter set, the machine can be operate either as a salient or non-salient pole rotor. The circuit nodes A – B – C are the terminals of star-connected stator winding. The component cannot be used with AC and DC simulation.

If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the synchronous machine are of special interest, voltmeters or ammeters can be connected to the synchronous machine.

In the case of identical parameters for the inductances L1D and L1Q, a synchronous machine with permanent magnet excitation and non-salient pole rotor is modeled. To model a permanent magnet salient-pole machine, the parameters must be different for L1D and L1Q.See also “Model Limits of Synchronous Machine Models” on page 170.

Equation System

The equation system is implemented in a rotor-fixed (rotor-fixed and also rotor flux-fixed) coordinate system (d-q-coordinates). Index 1 represents the stator quantities. The phase quantities of the real three-phase synchronous machine are indicated with a, b, c.

Voltage equations

$$v_{1d}(t) = i_{1d}(t) \cdot R_1 + \frac{d\psi_{1d}(t)}{dt} - p \cdot \omega(t) \cdot \psi_{1q}(t)$$

$$v_{1q}(t) = i_{1q}(t) \cdot R_1 + \frac{d\psi_{1q}(t)}{dt} + p \cdot \omega(t) \cdot \psi_{1d}(t)$$

Flux-linkage equations

$$\psi_{1d}(t) = i_{1d}(t) \cdot L_{1d} + k_e$$

$$\psi_{1q}(t) = i_{1q}(t) \cdot L_{1q}$$

Torque equation (electromagnetic developed "internal" torque)

$$m_i(t) = \frac{3}{2} p \cdot (\psi_{1d}(t) \cdot i_{1q}(t) - \psi_{1q}(t) \cdot i_{1d}(t))$$

Motion equation

$$\frac{d\omega(t)}{dt} = \frac{1}{J}(m_i(t) - m_w(t))$$

Electrical coupling of phase quantities with the orthogonal quantities

$$\gamma(t) = p \cdot \int \omega(t)dt$$

$$v_{1\alpha}(t) = \frac{2}{3}v_{1a}(t) - \frac{1}{3}v_{1b}(t) - \frac{1}{3}v_{1c}(t)$$

$$v_{1\beta}(t) = \frac{\sqrt{3}}{3}v_{1b}(t) - \frac{\sqrt{3}}{3}v_{1c}(t)$$

$$v_{1d}(t) = v_{1\alpha}(t) \cdot \cos(\gamma(t)) + v_{1\beta}(t) \cdot \sin(\gamma(t))$$

$$v_{1q}(t) = v_{1\alpha}(t) \cdot \sin(\gamma(t)) - v_{1\beta} \cdot \cos\gamma(t)$$

$$i_{1\alpha}(t) = i_{1d}(t) \cdot \cos(\gamma(t)) - i_{1q}(t) \cdot \sin(\gamma(t))$$

$$i_{1\beta}(t) = i_{1d}(t) \cdot \sin(\gamma(t)) + i_{1q}(t) \cdot \cos(\gamma(t))$$

$$i_{1a}(t) = i_{1\alpha}(t)$$

$$i_{1b}(t) = -\frac{1}{2}i_{1\alpha}(t) + \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

$$i_{1c}(t) = -\frac{1}{2}i_{1\alpha}(t) - \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

Dialog Settings	
↓↑	«Parameters» Click the «Value» field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
↓↑	«Initial Values» Click the «Value» field of the corresponding parameter to define the initial values. Common parameter types. Enter a numerical value, a variable, or an expression. The values are set only once at simulation start. Look the meaning up in the parameter table below.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension. The dimension is valid for all angle input quantities in the machine model.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Load Torque [Nm]	LOAD	real
Stator Resistance [Ω]	R1	real
Stator Inductance d-Axis [H]	L1D	real
Stator Inductance q-Axis [H]	L1Q	real
Rotor Flux-linkage [Vs]	KE	real
Number of Pole Pairs [/]	P	real
Moment of Inertia [kg*m²]	J	real
Initial Current Stator Phase a [A]	I1A0	real
Initial Current Stator Phase b [A]	I1B0	real
Initial Current Stator Phase c [A]	I1C0	real
Initial Rotor Speed [rpm]	N0	real
Initial Rotor Position [deg]	PHI0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Rotor Speed [rpm]	N	real
Inner Torque [Nm]	MI	real

Rotor rational Speed [1/s]	OMEGA	real
Angular Rotor Acceleration [1/s²]	ALPHA	real
Rotor Angle in Radians [rad]	PHI	real
Rotor Angle in Degrees [deg]	PHIDEG	real
Rotor Angle Mechanical in Radians [rad]	PHIM	real
Rotor Angle Mechanical in Degrees [deg]	PHIMDEG	real
Current Stator Phase a [A]	I1A	real
Current Stator Phase b [A]	I1B	real
Current Stator Phase c [A]	I1C	real
Real Part of Stator Current [A]	I1D	real
Imaginary Part of Stator Current [A]	I1Q	real
Real Part of Stator Voltage [V]	V1D	real
Imaginary Part of Stator Voltage [V]	V1Q	real
Real Part of Stator Flux Linkage [Vs]	PSI1D	real
Imaginary Part of Stator Flux Linkage [Vs]	PSI1Q	real

Component Nodes

Description	Node Name	Nature
Stator Node A	A	electrical
Stator Node B	B	electrical
Stator Node C	C	electrical

Synchronous Machine Permanent Excitation with Damper

>>Basics>Circuit>Electrical Machines

The model represents a permanent excitation synchronous machine with damper circuit i as a lumped circuit component. The damper circuit is not accessible from outside the machine; the corresponding phase windings are connected at a virtual neutral node. Depending on the parameter set, the machine can be operated either as a salient or non-salient pole rotor. The circuit nodes A – B – C are the terminals of star-connected stator winding. The component cannot be used with AC and DC simulation.

If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the synchronous machine are of special interest, voltmeters or ammeters can be connected to the synchronous machine.

In the case of identical parameters for the inductances L1D and L1Q, a synchronous machine with permanent magnet and non-salient pole rotor is modeled. To model a permanent magnet salient-pole machine, the parameters must be different for L1D and L1Q. See also “Model Limits of Synchronous Machine Models” on page 170.

Equation System

The equation system is implemented in a rotor-fixed (rotor-fixed and also rotor flux-fixed) coordinate system (d-q-coordinates). Index *1* represents the stator quantities; the index *2* the equivalent quantities of the damper circuit belonged to the rotor. The phase quantities of the real three-phase synchronous machine are indicated with *a*, *b*, *c*.

Voltage equations

$$v_{1d}(t) = i_{1d}(t) \cdot R_1 + \frac{d\psi_{1d}(t)}{dt} - p \cdot \omega(t) \cdot \psi_{1q}(t) \qquad 0 = i_{2d}(t) \cdot R_2 + \frac{d\psi_{2d}(t)}{dt}$$

$$v_{1q}(t) = i_{1q}(t) \cdot R_1 + \frac{d\psi_{1q}(t)}{dt} + p \cdot \omega(t) \cdot \psi_{1d}(t) \qquad 0 = i_{2q}(t) \cdot R_2 + \frac{d\psi_{2q}(t)}{dt}$$

Flux-linkage equations

$$\psi_{1d}(t) = i_{1d}(t) \cdot L_{1d} + i_{2d}(t) \cdot L_{m12d} + k_e$$
$$\psi_{1q}(t) = i_{1q}(t) \cdot L_{1q} + i_{2d}(t) \cdot L_{m12q}$$
$$\psi_{2d}(t) = i_{1d}(t) \cdot L_{m12d} + i_{2d}(t) \cdot L_{2d} + k_e$$
$$\psi_{2q}(t) = i_{1q}(t) \cdot L_{m12q} + i_{2q}(t) \cdot L_{2q}$$

Torque equation

$$m_i(t) = \frac{3}{2}p \cdot (\psi_{1d}(t) \cdot i_{1q}(t) - \psi_{1q}(t) \cdot i_{1d}(t))$$

Motion equation

$$\frac{d\omega(t)}{dt} = \frac{1}{J}(m_i(t) - m_w(t))$$

Electrical coupling of phase quantities with the orthogonal quantities

$$\gamma(t) = p \cdot \int \omega(t)dt$$
$$v_{1\alpha}(t) = \frac{2}{3}v_{1a}(t) - \frac{1}{3}v_{1b}(t) - \frac{1}{3}v_{1c}(t)$$
$$v_{1\beta}(t) = \frac{\sqrt{3}}{3}v_{1b}(t) - \frac{\sqrt{3}}{3}v_{1c}(t)$$
$$v_{1d}(t) = v_{1\alpha}(t) \cdot \cos(\gamma(t)) + v_{1\beta}(t) \cdot \sin(\gamma(t))$$
$$v_{1q}(t) = v_{1\alpha}(t) \cdot \sin(\gamma(t)) - v_{1\beta} \cdot \cos\gamma(t)$$

$$i_{1\alpha}(t) = i_{1d}(t) \cdot \cos(\gamma(t)) - i_{1q}(t) \cdot \sin(\gamma(t))$$
$$i_{1\beta}(t) = i_{1d}(t) \cdot \sin(\gamma(t)) + i_{1q}(t) \cdot \cos(\gamma(t))$$
$$i_{1a}(t) = i_{1\alpha}(t)$$
$$i_{1b}(t) = -\frac{1}{2}i_{1\alpha}(t) + \frac{\sqrt{3}}{2}i_{1\beta}(t)$$
$$i_{1c}(t) = -\frac{1}{2}i_{1\alpha}(t) - \frac{\sqrt{3}}{2}i_{1\beta}(t)$$

Dialog Settings	
	«Parameters» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
	«Initial Values» Click the **«Value»** field of the corresponding parameter to define the initial values. Common parameter types. Enter a numerical value, a variable, or an expression. The values are set only once at simulation start. Look the meaning up in the parameter table below.
	«Angular Dimension» In the list, select radians or degrees for the angular dimension. The dimension is valid for all angle input quantities in the machine model.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Load Torque [Nm]	LOAD	real
Stator Resistance [Ω]	R1	real

Damper Resistance [Ω]	R2	real
Stator Inductance d-Axis [H] Leakage inductance of stator side	L1D	real
Stator Inductance q-Axis [H] Leakage inductance of stator side	L1Q	real
Damper Inductance d-Axis [H] Leakage inductance of damper circuit side	L2D	real
Damper Inductance q-Axis [H] Leakage inductance of damper circuit side	L2Q	real
Mutual Inductance stator-damper d-Axis [H]	LM12D	real
Mutual Inductance stator-damper q-Axis [H]	LM12Q	real
Rotor Flux Linkage [Vs]	KE	real
Number of Pole Pairs [/]	P	real
Moment of Inertia [kg*m^2]	J	real
Initial Current Stator Phase a [A]	I1A0	real
Initial Current Stator Phase b [A]	I1B0	real
Initial Current Stator Phase c [A]	I1C0	real
Initial Current Damper Phase a [A]	I2A0	real
Initial Current Damper Phase b [A]	I2B0	real
Initial Current Damper Phase c [A]	I2C0	real
Initial Rotor Speed [rpm]	N0	real
Initial Rotor Position [deg]	PHI0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Rotor Speed [rpm]	N	real
Inner Torque [Nm]	MI	real
Rotor rational Speed [1/s]	OMEGA	real
Angular Rotor Acceleration [1/s^2]	ALPHA	real
Rotor Angle in Radians [rad]	PHI	real
Rotor Angle in Degrees [deg]	PHIDEG	real
Rotor Angle Mechanical in Radians [rad]	PHIM	real
Rotor Angle Mechanical in Degrees [deg]	PHIMDEG	real
Current Stator Phase a [A]	I1A	real
Current Stator Phase b [A]	I1B	real
Current Stator Phase c [A]	I1C	real
Current Damper Phase a [A]	I2A	real
Current Damper Phase b [A]	I2B	real
Current Damper Phase c [A]	I2C	real
Real Part of Stator Current [A]	I1D	real
Imaginary Part of Stator Current [A]	I1Q	real
Real Part of Damper Circuit Current [A]	I2D	real
Imaginary Part of Damper Circuit Current [A]	I2Q	real
Real Part of Stator Voltage [V]	V1D	real

Imaginary Part of Stator Voltage [V]	V1Q	real
Real Part of Stator Flux Linkage [Vs]	PSI1D	real
Imaginary Part of Stator Flux Linkage [Vs]	PSI1Q	real
Real Part of Damper Circuit Flux Linkage [Vs]	PSI2D	real
Imaginary Part of Damper Circuit Flux Linkage [Vs]	PSI2Q	real

Component Nodes

Description	Node Name	Nature
Stator Node A	A	electrical
Stator Node B	B	electrical
Stator Node C	C	electrical

DC Machine Permanent Excitation

>>Basics>Circuit>Electrical Machines
The model represents a DC machine with permanent excitation as a lumped circuit component. The component cannot be used with AC and DC simulation.

Model Limits of DC Machine Models

- The nonlinear magnetic circuit (DC machine with Nonlinear electrical excitation) is able to consider the dependence on excitation flux and inductance caused by the excitation current.
- Armature and exciter circuit of the DC machine model are considered to be completely decoupled.
- No consideration of saturation effects in the armature q-axis caused by the armature current.
- No consideration of armature reaction on exciting field
- No consideration of eddy-current and hysteresis loss caused by armature rotation and pulsating-current supply system
- Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally

Equation System

The equation system is implemented on condition of a linear magnetic circuit. Index *a* represents the armature circuit quantities.

Voltage equation

$$v_a(t) = i_a(t) \cdot R_a + \frac{di_a(t)}{dt} \cdot L_a + k_e \cdot \omega(t)$$

Torque equation (electromagnetic developed "internal" torque)

$$m_i(t) = k_e \cdot i_a(t)$$

Motion equation

$$\frac{d\omega(t)}{dt} = \frac{1}{J}(m_i(t) - m_w(t))$$

4

Dialog Settings	
⭳⭱	«Parameters» Defines the machine parameters in the table. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. Look the parameter meaning up in the parameter table below.
⭳⭱	«Initial Values» Defines the initial values of the machine in the table. Enter a numerical value, a variable, or an expression in the text box to define the initial values at simulation start. Look the parameter meaning up in the parameter table below.
▼	«Angular Dimension» Select the angular dimension – radians or degrees – in the list. The dimension is valid for all angle input quantities in the machine model.

Component Parameters

Description [Unit]	Parameter Name	
Load Torque [Nm]	LOAD	real
Armature Resistance [Ω] Sum of resistances of all windings of armarture circuit	RA	real
Armature Inductance [H] Sum of inductivities of all windings of armarture circuit	LA	real
Rotor Flux Linkage [Vs]	KE	real
Moment of Inertia [kg*m^2]	J	real
Initial Armature Current [A]	IA0	real
Initial Rotor Speed [rpm]	N0	real
Initial Rotor Position [deg]	PHI0	real

Component Outputs

Description [Unit]	Parameter Name	
Rotor Speed [rpm]	N	real
Inner Torque [Nm]	MI	real
Rotor rational Speed [1/s]	OMEGA	real
Angular Rotor Acceleration [1/s^2]	ALPHA	real
Rotor Angle in Radians [rad]	PHI	real
Rotor Angle in Degrees [deg]	PHIDEG	real
Armature Voltage [V]	VA	real
Armature Current [A]	IA	real
Linked Armature Flux [Vs]	PSIA	real
1st derivative Armature Voltage [V/s]	dVA	real
1st derivative Armature Current [A/s]	dIA	real
1st derivative linked Armature Flux [V]	dPSIA	real

Component Nodes

Description	Node Name	Nature
Armature Node N1 (with red point) and N2	N1/N2	electrical

DC Machine Linear Electrical Excitation

>>Basics>Circuit>Electrical Machines

The model represents a DC machine as a lumped circuit component. By proper connection of armature and excitation circuit, separately excited, series and shunt machines can be modeled. The component cannot be used with AC and DC simulation. See also “Model Limits of DC Machine Models” on page 181.

Equation System

The equation system is implemented on condition of a linear magnetic circuit. Index ***a*** represents the armature circuit quantities, the index ***e*** the quantities of excitation circuit.

Voltage equations

$$v_a(t) = i_a(t) \cdot R_a + \frac{di_a(t)}{dt} \cdot L_a + i_e(t) \cdot IEPSI \cdot \omega(t) \qquad v_e(t) = i_e(t) \cdot R_e + \frac{di_e(t)}{dt} \cdot L_e$$

Flux-linkage equation

$$\Psi_e(t) = i_e(t) \cdot L_e$$

Torque equation (electromagnetic developed “internal” torque)

$$m_i(t) = i_e(t) \cdot IEPSI \cdot i_a(t)$$

Motion equation

$$\frac{d\omega(t)}{dt} = \frac{1}{J}(m_i(t) - m_w(t))$$

Dialog Settings	
↓↑	«Parameters» Click the «Value» field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
↓↑	«Initial Values» Click the «Value» field of the corresponding parameter to define the initial values. Common parameter types. Enter a numerical value, a variable, or an expression. The values are set only once at simulation start. Look the meaning up in the parameter table below.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension. The dimension is valid for all angle input quantities in the machine model.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Load Torque [Nm]	LOAD	real
Armature Resistance [Ω] Sum of resistances of all windings of armarture circuit	RA	real
Armature Inductance [H] Sum of inductivities of all windings of armarture circuit	LA	real
Excitation Resistance [Ω] Sum of resistances of all windings of excitation circuit	RE	real
Excitation Inductance [H] Sum of inductivities of all windings of excitation circuit	LE	real
Moment of Inertia [kg*m²]	J	real

Excitation Current-Flux Rate [Vs/A] Nominal excitation flux-linkage normalized to nominal excitation current	IEPSI	real
Initial Armature Current [A]	IA0	real
Initial Excitation Current [A]	IE0	real
Initial Rotor Speed [rpm]	N0	real
Initial Rotor Position [deg]	PHI0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Rotor Speed [rpm]	N	real
Inner Torque [Nm]	MI	real
Rotor Rational Speed [1/s]	OMEGA	real
Angular Rotor Acceleration [$1/s^2$]	ALPHA	real
Rotor Angle in Radians [rad]	PHI	real
Rotor Angle in Degrees [deg]	PHIDEG	real
Armature Voltage [V]	VA	real
Armature Current [A]	IA	real
Excitation Current [A]	IE	real
Linked Armature Flux [Vs]	PSIA	real
1st derivative Armature Voltage [V/s]	dVA	real
1st derivative Armature Current [A/s]	dIA	real
1st derivative Excitation Current [A]	dIE	real
1st derivative linked Armature Flux [V]	dPSIA	real

Component Nodes

Description	Node Name	Nature
Armature Circuit Node N1 (with red point)	N1	electrical
Armature Circuit Node N2	N2	electrical
Excitation Circuit Node E1	E1	electrical
Excitation Circuit Node E2	E2	electrical

DC Machine Nonlinear Electrical Excitation

>>Basics>Circuit>Electrical Machines

The model represents a DC machine as a lumped circuit component. By proper connection of armature and excitation circuit, separately excited, series and shunt machines can be modeled. The component cannot be used with AC and DC simulation. See also “Model Limits of DC Machine Models” on page 181.

Equation System

The equation system is implemented on condition of a nonlinear magnetic circuit (saturation) and the resulting nonlinear excitation circuit inductivity. Index ***a*** represents the armature circuit quantities, the index ***e*** the quantities of excitation circuit.

Voltage equations

$$v_a(t) = i_a(t) \cdot R_a + \frac{di_a(t)}{dt} \cdot L_a + \Psi(i_e) \cdot \omega(t) \qquad v_e(t) = i_e(t) \cdot R_e + \frac{di_e(t)}{dt} \cdot L_e(i_e)$$

Torque equation (electromagnetic developed "internal" torque)

$$m_i(t) = \Psi(i_e) \cdot i_a(t)$$

Motion equation

$$\frac{d\omega(t)}{dt} = \frac{1}{J}(m_i(t) - m_w(t))$$

Dialog Settings	
↓↑	«Parameters» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
↓↑	«Initial Values» Click the **«Value»** field of the corresponding parameter to define the initial values. Common parameter types. Enter a numerical value, a variable, or an expression. The values are set only once at simulation start. Look the meaning up in the parameter table below.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension. The dimension is valid for all angle input quantities in the machine model.
	«Excitation Inductance Characteristic» Defines the machine characteristic L=f(i). Open the characteristic dialog or use the pin to define a characteristic. See also “Separate Component Characteristic” on page 297.
	«Magnetization Characteristic» Defines the machine characteristic PSI=f(i). Open the characteristic dialog or use the pin to define a characteristic. See also “Separate Component Characteristic” on page 297.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Load Torque [Nm]	LOAD	real
Armature Resistance [Ω] Sum of resistances of all windings of armarture circuit	RA	real

Armature Inductance [H] Sum of inductivities of all windings of armarture circuit	LA	real
Excitation Resistance [Ω] Sum of resistances of all windings of excitation circuit	RE	real
Moment of Inertia [kg*m²]	J	real
Initial Armature Current [A]	IA0	real
Initial Excitation Current [A]	IE0	real
Initial Rotor Speed [rpm]	N0	real
Initial Rotor Position [rad]	PHI0	real
Excitation Inductance Characteristic Sum of inductivities of all windings of excitation circuit $L_e=f(i_e)$	CHLE	real
Magnetization Characteristic Nonlinear characteristic of excitation flux-linkage $\Psi=f(i_e)$	CHM	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Rotor Speed [rpm]	N	real
Inner Torque [Nm]	MI	real
Rotor rational Speed [1/s]	OMEGA	real
Angular Rotor Acceleration [1/s²]	ALPHA	real
Rotor Angle in Radians [rad]	PHI	real
Rotor Angle in Degrees [deg]	PHIDEG	real
Armature Voltage [V]	VA	real
Armature Current [A]	IA	real
Excitation Current [A]	IE	real
Linked Armature Flux [Vs]	PSIA	real
1st derivative Armature Voltage [V/s]	dVA	real
1st derivative Armature Current [A/s]	dIA	real
1st derivative Excitation Current [A]	dIE	real
1st derivative linked Armature Flux [V]	dPSIA	real

Component Nodes

Description	Node Name	Nature
Armature Circuit Node N1 (with red point)	N1	electrical
Armature Circuit Node N2	N2	electrical
Excitation Circuit Node E1	E1	electrical
Excitation Circuit Node E2	E2	electrical

4.8 Transformers

- Single-Phase Systems
- Three-Phase Systems

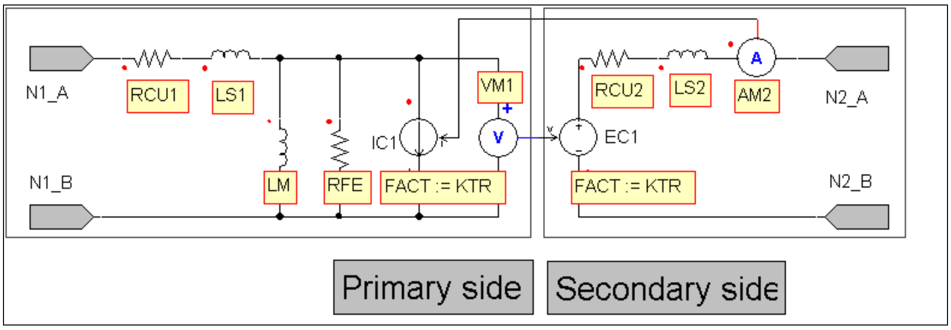
SIMPLORER Transformer Models According to Spiro

Transformer components, available in the Circuit folder of the «**Basics**» tab, are based on the *Spiro* model, which is characterized by any energy flow direction, a free number of galvanically separate windings, the exclusion of DC-current transmission and the consideration of losses.

In the Spiro model, every limb of the transformer is represented by one inductance. Each winding, that influences the flux through the limb, contributes to the current flowing through the inductance. Usually the ratio flux/current is different for the winding of the limb because of different numbers of turns or, in the case of a three-phase transformers, different positions of the windings. The current of only one winding can flow directly through the inductance. The other windings contribute by coupling factors.

Single Phase

The circuit shows the equivalent circuit for a single phase transformer. The current through the primary winding contributes directly to the current through the main inductance. Therefore the main inductance is calculated with respect to the primary winding.



Because of primary and secondary side stray inductances coupling factors are introduced. The coupling factor for the secondary side is determined by the ratio of the number of turns only. Therefore the current through the secondary winding controls the current source at the primary side with the control factor KTR which is the number-of-turns ratio.

The same relationships apply for the voltage induced at primary and secondary side respectively. Since the main inductance is calculated for the primary winding and its current is calculated with respect to the primary side, the induced voltage at the primary side equals the voltage across the main inductance. The induced voltage at the secondary side equals the voltage across the main inductance multiplied by the number-of-turns ratio KTR.

Model Extension for Three-Phase Systems

To model a three-phase system, the coupling between all six windings has to be considered. The model for the single-phase transformer has to be used three times. For every phase the influence of two more primary windings and two more secondary windings has to be modeled. This influence is modeled as for the single-phase transformer using controlled current and voltage sources.

Assuming the same number of turns for all primary windings the coupling between the primary windings depends on the geometry only. The coupling of a secondary winding to the primary winding of the same phase depends on the number-of-turns ratio again (as for single-phase transformers).The coupling from a secondary side to a primary side of another phase depends on geometry and number-of-turns ratio.

Three-Phase Transformer SIMPLORER Models

SIMPLROER provides a separate models for primary and secondary windings, which can be adjusted to your needs individually. The coupling factors can be determined by a FEM-analysis or by an estimation of the magnetic resistances based on the core geometry.

There are also two parameterized models for three-phase transformers of *small* and *large* power. The parameters were determined by analyzing existing transformers with so-called EI-core. See also “Six-winding Transformer (small and large power)” on page 194.

See also Spiro, Hans: *Simulation integrierter Schaltungen durch universelle Rechenprogramme*,Verfahren und Praxis der rechnergestützten Simulation nichtlinearer Schaltungen, R. Oldenbourg Verlag München Wien, 1985

4.8.1 Single-Phase Systems

- Ideal Two-winding Transformer (TWT)
- Linear Two-winding Transformer
- Nonlinear Two-winding-Transformer
- Primary Side of a Two-winding-Transformer
- Secondary Side of a Two-winding-Transformer

Ideal Two-winding Transformer

>>Basics>Circuit>Transformers>Single-Phase Systems

The component represents the mutual coupling between two inductors, L1 and L2, quantitatively determined by the factor k. The ideal two-winding transformer is an internal component which is not based on the *Spiro* model.

$$M = K \cdot \sqrt{L_1 \cdot L_2} \qquad \text{with} \qquad K = K_1 \cdot K_2$$

Dialog Settings
«Inductance L1/L2 [H]» L1 and L2 are the values of coupled inductances. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. You can also specify an initial current for each inductance.
«Initial Current L1/L2 [A]» I01 and I02 are the values of the initial currents. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. The values are set only once at simulation start.
«Mutual Inductance» K/M is the factor which quantitatively determines the mutual inductances. Common parameter types. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.

☑ ☐	«Use pin» If the box is checked, the coupling factor is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Inductance L1 [H]	L1	real
Initial Current L1 [A]	I01	real
Inductance L2 [H]	L2	real
Initial Current L2 [A]	I02	real
Coupling Factor [/]	K	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage L1 [V]	V1	real
Voltage L2 [V]	V2	real
Current L1 [A]	I1	real
Current L2 [A]	I2	real
Flux Linkage L1 [Vs]	PSI1	real
Flux Linkage L2 [Vs]	PSI2	real
Mutual Inductance [H]	M	real
Derivative of Voltage L1 [V/s]	dV1	real
Derivative of Voltage L2 [V/s]	dV2	real
Derivative of Current [A/s]	dI1	real
Derivative of Current [A/s]	dI2	real
Derivative of Flux Linkage L1 [V]	dPSI1	real
Derivative of Flux Linkage L2 [V]	dPSI2	real

Component Nodes

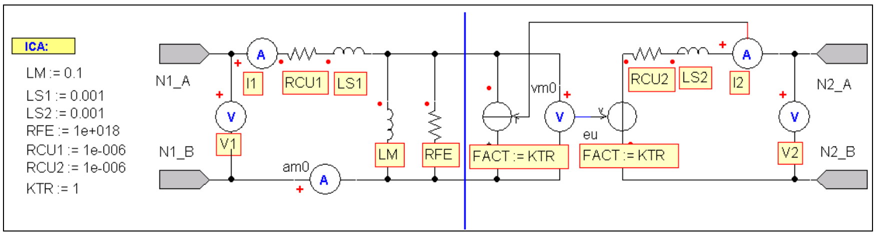
Description	Node Name	Nature
Inductor 1 – Node N1 and N2	N1L1, N2L1	electrical
Inductor 2 – Node N1 and N2	N1L2, N2L2	electrical

Linear Two-winding Transformer

>>Basics>Circuit>Transformers>Single-Phase Systems

The component represents a transformer based on the *Spiro* model. For each coil make sure a network path exists to the ground node. The macro does not permit an access to the initial value of currents, these are assumed to be zero. Within the dialog you can define the main and leakage inductances, the equivalent resistances for iron losses and the winding resistances for each side.

$$KTR = \frac{W_2}{W_1} \quad \text{with} \quad W_1 = \text{Windings Primary Side} \quad \text{and} \quad W_2 = \text{Windings Secondary Side}$$



Dialog Settings	
	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
	«Name»«Unit»«Description» Displays the parameter names and their corresponding unit and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter use the **«Info»** field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Main Inductance [H]	LM	real
Leakage Inductance Primary Side [H]	LS1	real
Leakage Inductance Secondary Side [H]	LS2	real
Equivalent Resistance for Iron Losses [Ω]	RFE	real
Winding Resistance Primary Side [Ω]	RCU1	real
Winding Resistance Secondary Side [Ω]	RCU2	real
Coupling Factor [/]	KTR	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage Primary Side [V]	V1	real
Voltage Secondary Side [V]	V2	real
Current Primary Side [A]	I1	real
Current Secondary Side [A]	I2	real

Component Nodes

Description	Node Name	Nature
Primary Side Node A and B	N1_A, N1_B	electrical
Secondary Side Node A and B	N2_A, N2_B	electrical

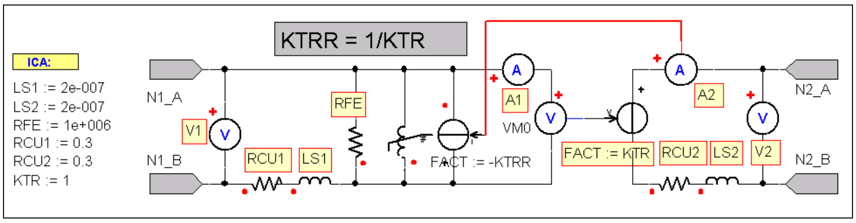
Nonlinear Two-winding Transformer

>>Basics>Circuit>Transformers>Single-Phase Systems

The component represents a transformer based on the *Spiro* model. The characteristic describes the nonlinear mutual inductance without hysteresis (L=f(I)). L has to be >0 within the characteristic.

For each coil make sure a network path exists to the ground node. The macro does not provide an access to the initial value of currents, these are assumed to be zero. Within the dialog you can define the main and leakage inductances, the equivalent resistances for iron losses and the winding resistances for each side.

$$KTR = \frac{W_2}{W_1} \quad \text{with} \quad \begin{array}{l} W_1 = \text{Windings Primary Side} \quad \text{and} \\ W_2 = \text{Windings Secondary Side} \end{array}$$



Dialog Settings

«Value»
Click the «Value» field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.

«Name»«Unit»«Description»
Displays the parameter names and their corresponding unit and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Main Inductance [H]	LM	real
Leakage Inductance Primary Side [H]	LS1	real
Leakage Inductance Secondary Side [H]	LS2	real
Equivalent Resistance for Iron Losses [Ω]	RFE	real
Winding Resistance Primary Side [Ω]	RCU1	real
Winding Resistance Secondary Side [Ω]	RCU2	real
Coupling Factor [/]	KTR	real
Characteristic	CH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage Primary Side [V]	V1	real
Voltage Secondary Side [V]	V2	real
Current Primary Side [A]	I1	real
Current Secondary Side [A]	I2	real

Component Nodes

Description	Node Name	Nature
Primary Side Node A and B	N1_A, N1_B	electrical
Secondary Side Node A and B	N2_A, N2_B	electrical

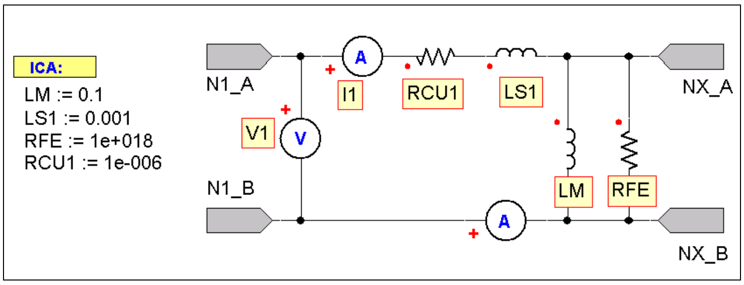
Primary Side of a Two-winding Transformer

>>Basics>Circuit>Transformers>Single-Phase Systems

The component represents a transformer based on the *Spiro* model. The use of this macro makes only sense with at least one macro of the secondary side. Using the Primary Side and Secondary Side of a Two-winding-Transformer allows a simple construction of a Single-Phase Transformer with more windings.

The terms primary and secondary are not to understand in the strict sense. It means only to combine exactly one macro of the primary side with at least one macro of the secondary side.

For each coil make sure a network path exists to the ground node. The macro does not provide an access to the initial value of currents, these are assumed to be zero. Within the dialog you can define the leakage inductances, the equivalent resistances for iron losses and the winding resistances for the primary side.



Dialog Settings	
	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
	«Name»«Unit»«Description» Displays the parameter names and their corresponding unit and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter use the **«Info»** field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Main Inductance [H]	LM	real
Leakage Inductance Primary Side [H]	LS1	real
Equivalent Resistance for Iron Losses [Ω]	RFE	real
Winding Resistance Primary Side [Ω]	RCU1	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage Primary Side [V]	V1	real
Current Primary Side [A]	I1	real

Component Nodes

Description	Node Name	Nature
Primary Side Node A and B	N1_A, N1_B	electrical
Open Side Node A and B	NX_A, NX_B	electrical

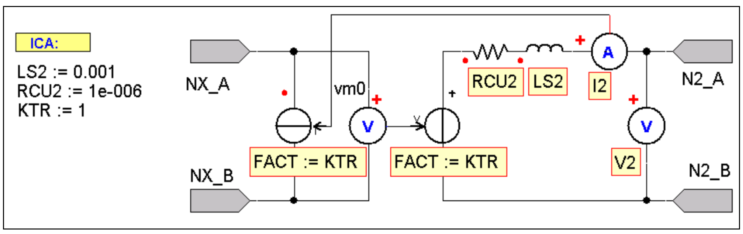
Secondary Side of a Two-winding Transformer

>>Basics>Circuit>Transformers>Single-Phase Systems

The component represents a transformer based on the *Spiro* model. The use of this macro makes only sense with at least one macro of the primary side. Using the Primary Side and Secondary Side of a Two-winding-Transformer allows a simple construction of a Single-Phase Transformer with more windings.

The terms primary and secondary are not to understand in the strict sense. It means only to combine exactly one macro of the primary side with at least one macro of the secondary side.

For each coil make sure a network path exists to the ground node. The macro does not provide an access to the initial value of currents, these are assumed to be zero. Within the dialog you can define the leakage inductance, the equivalent resistance for iron losses and the winding resistance for the secondary side.



Dialog Settings	
	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.

«Name»«Unit»«Description»
Displays the parameter names and their corresponding unit and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter use the **«Info»** field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Leakage Inductance Secondary Side [H]	LS2	real
Winding Resistance Secondary Side [Ω]	RCU2	real
Coupling Factor	KTR	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage Secondary Side [V]	V2	real
Current Secondary Side [A]	I2	real

Component Nodes

Description	Node Name	Nature
Open Side Node A and B	NX_A, NX_B	electrical
Secondary Side Node A and B	N2_A, N2_B	electrical

4.8.2 Three-Phase Systems

- Six-winding Transformer (small power)
- Six-winding Transformer (large power)
- Primary Side of a Six-winding-Transformer
- Secondary Side of a Six-winding-Transformer

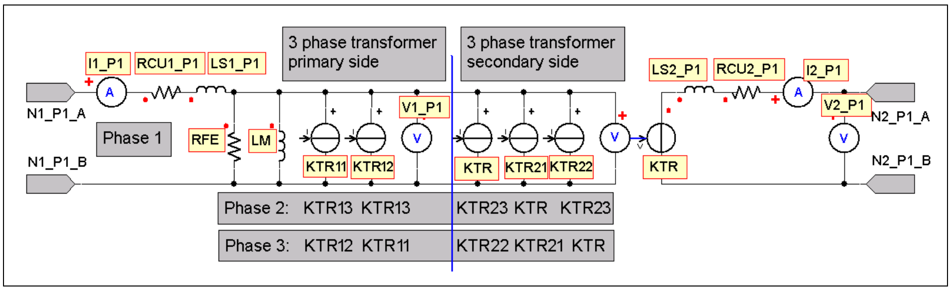
Six-winding Transformer (small and large power)

>>Basics>Circuit>Transformers>Three-Phase Systems
The component represents a transformer based on the *Spiro* model. For each coil make sure a network path exists to the ground node. The macro does not provide an access to the initial value of currents, these are assumed to be zero. Within the dialog you can define the main and leakage inductances, the equivalent resistances for iron losses and the winding resistances for each phase and side.

$$KTR = \frac{W_2}{W_1} \quad \text{with} \quad \begin{array}{l} W_1 = \text{Windings Primary Side} \quad \text{and} \\ W_2 = \text{Windings Secondary Side} \end{array}$$

Small Power
KTR11=0.5, KTR12=0.25, KTR13=0.75,
KTR21=0.5*KTR, KTR22=0.25*KTR, KTR23=0.75*KTR

Large Power
KTR11=0.5, KTR12=0.333, KTR13=0.667,
KTR21=0.5*KTR, KTR22=0.333*KTR, KTR23=0.667*KTR



Dialog Settings

«Value»
Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.

«Name»«Unit»«Description»
Displays the parameter names and their corresponding unit and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter use the **«Info»** field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Main Inductance [H]	LM	real
Leakage Inductance Primary Side Phase 1 [H]	LS1_P1	real
Leakage Inductance Primary Side Phase 2 [H]	LS1_P2	real
Leakage Inductance Primary Side Phase 3 [H]	LS1_P3	real
Leakage Inductance Secondary Side Phase 1 [H]	LS2_P1	real
Leakage Inductance Secondary Side Phase 2 [H]	LS2_P2	real
Leakage Inductance Secondary Side Phase 3 [H]	LS2_P3	real
Equivalent Resistance for Iron Losses [Ω]	RFE	real
Winding Resistance Primary Side Phase 1 [Ω]	RCU1_P1	real
Winding Resistance Primary Side Phase 2 [Ω]	RCU1_P2	real
Winding Resistance Primary Side Phase 3 [Ω]	RCU1_P3	real
Winding Resistance Secondary Side Phase 1 [Ω]	RCU2_P1	real

Winding Resistance Secondary Side Phase 2 [Ω]	RCU2_P2	real
Winding Resistance Secondary Side Phase 3 [Ω]	RCU2_P3	real
Coupling Factor [/]	KTR	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage Primary Side Phase 1 [V]	V1_P1	real
Voltage Primary Side Phase 2 [V]	V1_P2	real
Voltage Primary Side Phase 3 [V]	V1_P3	real
Voltage Secondary Side Phase 1 [V]	V2_P1	real
Voltage Secondary Side Phase 2 [V]	V2_P2	real
Voltage Secondary Side Phase 3 [V]	V2_P3	real
Current Primary Side Phase 1 [A]	I1_P1	real
Current Primary Side Phase 2 [A]	I1_P2	real
Current Primary Side Phase 3 [A]	I1_P3	real
Current Secondary Side Phase 1 [A]	I2_P1	real
Current Secondary Side Phase 2 [A]	I2_P2	real
Current Secondary Side Phase 3 [A]	I2_P3	real

Component Nodes

Description	Node Name	Nature
Primary Side Phase 1 Node A and B	N1_P1_A, N1_P1_B	electrical
Secondary Side Phase 1 Node A and B	N2_P1_A, N2_P1_B	electrical
Primary Side Phase 2 Node A and B	N1_P2_A, N1_P2_B	electrical
Secondary Side Phase 2 Node A and B	N2_P2_A, N2_P2_B	electrical
Primary Side Phase 3 Node A and B	N1_P3_A, N1_P3_B	electrical
Secondary Side Phase 3 Node A and B	N2_P3_A, N2_P3_B	electrical

Primary Side of a Six-winding Transformer

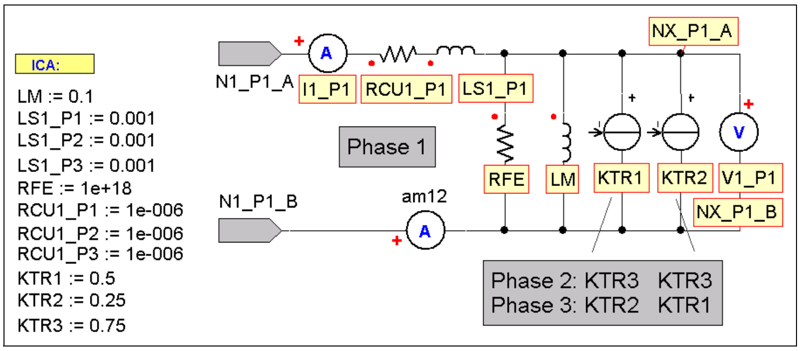
>>Basics>Circuit>Transformers>Three-Phase Systems

The transformer model is based on the *Spiro* model.The use of this macro makes only sense with at least one macro of the secondary side. Using the Primary Side and Secondary Side of a Six-winding-Transformer allows a simple construction of a Three-Phase Transformer with more windings.

The terms primary and secondary are not to understand in the strict sense. It means only to combine exactly one macro of the primary side with at least one macro of the secondary side.

For each coil make sure a network path exists to the ground node. The macro does not provide an access to the initial value of currents, these are assumed to be zero. Within the dialog you can define the main and leakage inductances, the equivalent resistances for iron losses and the winding resistances for each phase.

$$KTR = \frac{W_2}{W_1} \quad \text{with} \quad \begin{aligned} W_1 &= \text{Windings Primary Side} \quad \text{and} \\ W_2 &= \text{Windings Secondary Side} \end{aligned}$$



Dialog Settings

«Value»
Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.

«Name»«Unit»«Description»
Displays the parameter names and their corresponding unit and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter use the **«Info»** field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Main Inductance [H]	LM	real
Leakage Inductance Primary Side Phase 1 [H]	LS1_P1	real
Leakage Inductance Primary Side Phase 2 [H]	LS1_P2	real
Leakage Inductance Primary Side Phase 3 [H]	LS1_P3	real
Equivalent Resistance for Iron Losses [Ω]	RFE	real
Winding Resistance Primary Side Phase 1 [Ω]	RCU1_P1	real
Winding Resistance Primary Side Phase 2 [Ω]	RCU1_P2	real
Winding Resistance Primary Side Phase 3 [Ω]	RCU1_P3	real
Coupling Factor 1/Coupling Factor 2/Coupling Factor 3 [/]	KTR1/KTR2/KTR3	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage Primary Side Phase 1 [V]	V1_P1	real
Voltage Primary Side Phase 2 [V]	V1_P2	real
Voltage Primary Side Phase 3 [V]	V1_P3	real
Current Primary Side Phase 1 [A]	I1_P1	real

Current Primary Side Phase 2 [A]	I1_P2	real
Current Primary Side Phase 3 [A]	I1_P3	real

Component Nodes

Description	Node Name	Nature
Primary Side Phase 1 Node A and B	N1_P1_A, N1_P1_B	electrical
Open Side Phase 1 Node A and B	NX_P1_A, NX_P1_B	electrical
Primary Side Phase 2 Node A and B	N1_P2_A, N1_P2_B	electrical
Open Side Phase 2 Node A and B	NX_P2_A, NX_P2_B	electrical
Primary Side Phase 3 Node A and B	N1_P3_A, N1_P3_B	electrical
Open Side Phase 3 Node A and B	NX_P3_A, NX_P3_B	electrical

Secondary Side of a Six-winding Transformer

>>Basics>Circuit>Transformers>Three-Phase Systems

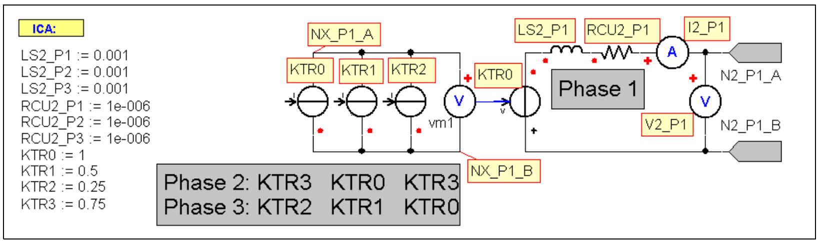
The component represents a transformer based on the *Spiro* model. The use of this macro makes only sense with at least one macro of the primary side. Using the Primary Side and Secondary Side of a Six-winding-Transformer allows a simple construction of a Three-Phase Transformer with more windings.

The terms primary and secondary are not to understand in the strict sense. It means only to combine exactly one macro of the primary side with at least one macro of the secondary side.

For each coil make sure a network path exists to the ground node. The macro does not provide an access to the initial value of currents, these are assumed to be zero. Within the dialog you can define the leakage inductance, the equivalent resistance for iron losses and the winding resistance for the secondary side.

For each coil make sure a network path exists to the ground node. The macro does not provide an access to the initial value of currents, these are assumed to be zero. Within the dialog you can define the main and leakage inductances, the equivalent resistances for iron losses and the winding resistances for each phase.

$$KTR = \frac{W_2}{W_1} \quad \text{with} \quad W_1 = \text{Windings Primary Side} \quad \text{and} \quad W_2 = \text{Windings Secondary Side}$$



Dialog Settings	
⭳⭱	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the meaning up in the parameter table below.
	«Name»«Unit»«Description» Displays the parameter names and their corresponding unit and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter use the **«Info»** field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Main Inductance [H]	LM	real
Leakage Inductance Primary Side Phase 1 [H]	LS1_P1	real
Leakage Inductance Primary Side Phase 2 [H]	LS1_P2	real
Leakage Inductance Primary Side Phase 3 [H]	LS1_P3	real
Equivalent Resistance for Iron Losses [Ω]	RFE	real
Winding Resistance Primary Side Phase 1 [Ω]	RCU1_P1	real
Winding Resistance Primary Side Phase 2 [Ω]	RCU1_P2	real
Winding Resistance Primary Side Phase 3 [Ω]	RCU1_P3	real
Coupling Factor 0 [/]	KTR0	real
Coupling Factor 1/Coupling Factor 2/Coupling Factor 3 [/]	KTR1/KTR2/KTR3	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage Secondary Side Phase 1 [V]	V2_P1	real
Voltage Secondary Side Phase 2 [V]	V2_P2	real
Voltage Secondary Side Phase 3 [V]	V2_P3	real
Current Secondary Side Phase 1 [A]	I2_P1	real
Current Secondary Side Phase 2 [A]	I2_P2	real
Current Secondary Side Phase 3 [A]	I2_P3	real

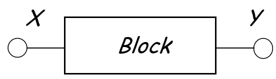
Component Nodes

Description	Node Name	Nature
Open Side Phase 1 Node A and B	NX_P1_A, NX_P1_B	electrical
Secondary Side Phase 1 Node A and B	N2_P1_A, N2_P1_B	electrical
Open Side Phase 2 Node A and B	NX_P2_A, NX_P2_B	electrical
Secondary Side Phase 2 Node A and B	N2_P2_A, N2_P2_B	electrical
Open Side Phase 3 Node A and B	NX_P3_A, NX_P3_B	electrical
Secondary Side Phase 3 Node A and B	N2_P3_A, N2_P3_B	electrical

5 Modeling with Block Diagrams

Transfer Blocks in SIMPLORER are defined by an internal identifier, the name and a set of parameters; Blocks have no conservative nodes. A block is a linear or nonlinear transfer element with a defined static or dynamic response characteristic. The block output signal does not influence the block input quantity (non-reacting).

Block components appear in the *Blocks* folder of the **«Basics»** tab. The blocks are subdivided into continuous, discrete, and source blocks as well as signal processing and mathematical blocks. Select the folder *Blocks*, open one of the folders, click a name, drag the component onto the sheet and release the mouse button.



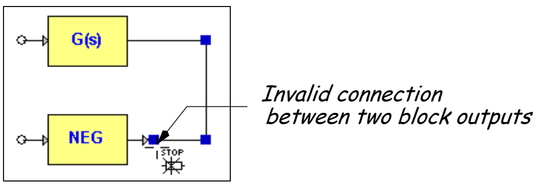
Non-reacting block

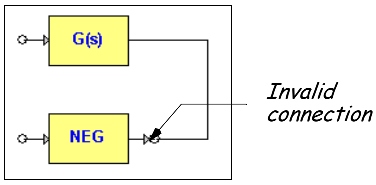
Block diagrams with typical transfer behaviors can be modeled for control and cybernetic systems by means of the available blocks.

The block output signal and the specific block dependent parameters can be used as outputs. The block output signal is represented by the «VAL» entry; the other parameters by their corresponding name.

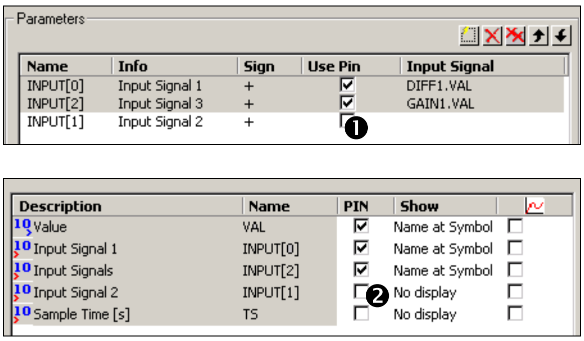
Connecting Transfer Blocks

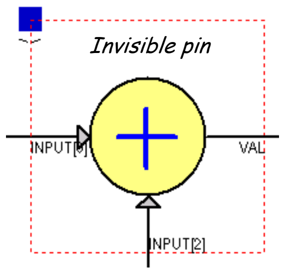
The structure of the block diagram is created in the Wire mode (CTRL+W or CONNECT>WIRE menu). Schematic tests the pins during the routing and will not allow invalid connections (e.g., between two inputs or two outputs).





If an input or an output in the block diagram is not connected, the **«Build»** Info Window displays a warning. To avoid this warning, you can hide any unused pins of an element. Open the property dialog and clear the check box of the pin you want to hide, either at the **«Parameters»** or **«Output/Display»** page. The pin is invisible on the sheet and the warning no longer occurs. See also 3.3.2 “Using Pins” on page 44.



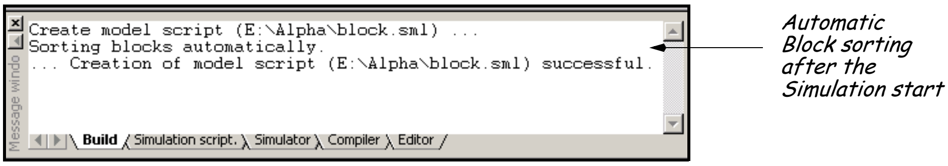


Signal Direction in Block Diagrams

The sequence of block computation can significantly influence the simulation result. You can change the block sequence with the SHEET>DETERMINE BLOCK SEQUENCE command.

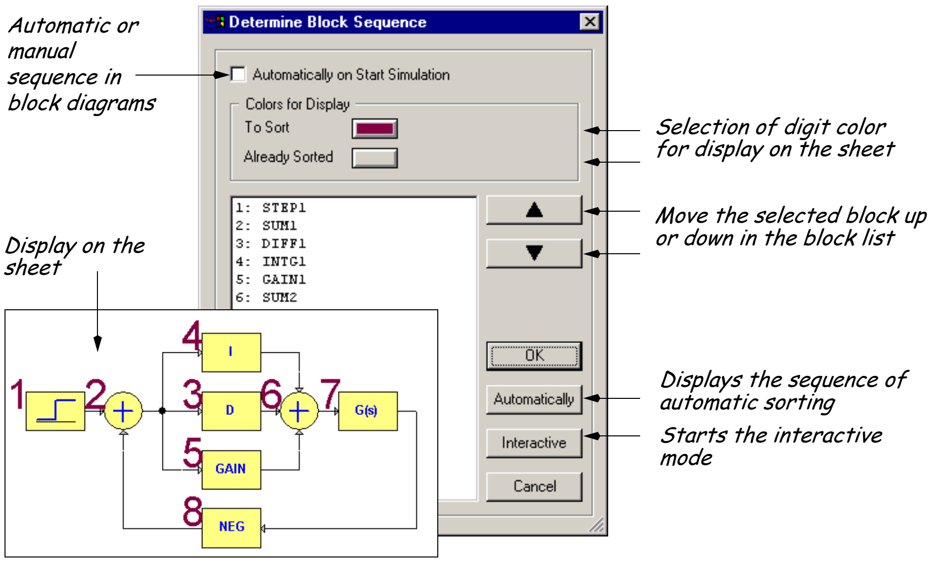
Using Automatic Block Sorting

When the automatic Block sorting mode in SHEET>DETERMINE BLOCK SEQUENCE is active, the blocks will be sorted after simulation start according to their signal direction. If the automatic block sorting is active, the «Build» Info Window displays a message after the simulation start.



Using Manual Block Sorting

To sort blocks manually, clear the option «automatically on start simulation» in SHEET>DETERMINE BLOCK SEQUENCE. Select <Interactive> and the current sequence is displayed in digits next to the blocks. The sequence can be changed by clicking the blocks in the desired sequence on the sheet or by arranging the block list in a new sequence in the dialog.



The following functions are active in the interactive mode:

- Click the block: Defines next position in the sequence.
- Double-click the block: Moves the block to the last position in the list of blocks already sorted.
- Double-click beside the element: Exits the mode and applies the settings.
- <ESC>: Exits the mode without applying the changes.

The block sequence in subsheets must be defined separately. Open the subsheet and choose SHEET>DETERMINE BLOCK SEQUENCE. The sequence starts in each subsheet with '1'.

Defining the Sample Time

You can choose system sample time '0' or a specific value for each block.

The sampling time must be greater than the minimum time step (`HMIN`) and less than the simulation end time (`TEND`). If the sampling time is defined with **«System»**, the blocks are calculated with the time step determined by the other modules (circuit, state graph, or equations), but at least by the maximum time step of integration.

5.1 Continuous Blocks

- S-Transfer Function (GS)
- Gain (GAIN)
- Derivative (DIFF)
- Integrator (INTG)
- Memory (DEAD)
- Dead-Time Element (DEAD)
- Delay (GZ)

S-Transfer Function

>>Basics>Blocks>Continuous Blocks

The Block represents a continuous transfer characteristic G(s) (Laplace-domain, Laplace-operator specified as **s**). Within the dialog you can define numerator and denominator order and coefficients of the S-Transfer Function.

$$G(s) = \frac{b_0 + b_1 s + b_2 s^2 + \ldots + b_n s^n}{a_0 + a_1 s + a_2 s^2 + \ldots + a_m s^m} \qquad a_m \neq 0$$

Dialog Settings
«Numerator»«Order» Enter a numerical value (non-negative integer with or without unit suffix) in the text box to define the numerator order of the transfer function. Please note: You cannot define a variable or expression in this case. Only a number is accepted. «Numerator»«Coefficient» Click the **«Value»** field of the corresponding numerator coefficient B[n] to define a value. Common parameter types. Enter a numerical value, a variable, or an expression.
«Denominator»«Order» Type a numerical value (non-negative integer with or without unit suffix) in the text box to define the denominator order of the transfer function. Please note: You cannot define a variable or expression in this case. Only a number is accepted.

	«Denominator»«Coefficient» Click the **«Value»** field of the corresponding denominator coefficient A[n] to define a value. Common parameter types. Enter a numerical value, a variable, or an expression.
▾	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Numerator Dimension	N	real
Denominator Dimension	D	real
Numerator Coefficients	B[n]	real
Denominator Coefficients	A[n]	real
Sum of Initial Values (only SML)	SUM0	real
Initial Values of D-components (only SML)	D0[n]	real
Initial Values of P-components (only SML)	I0[n]	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Gain

>>Basics>Blocks>Continuous Blocks

The block represents a proportional gain (Laplace-domain, Laplace-operator specified as **s**).

$$y(k) = K \cdot x(k) \qquad \text{or} \qquad G(s) = K$$

Dialog Settings	
	«Gain» Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.

☑ ☐	«Use Pin» If the box is checked, the gain is defined by the quantity connected to the pin, otherwise the value in the text box is used.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Proportional Gain	K	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Integrator

>>Basics>Blocks>Continuous Blocks

The block represents an integration block with limitation and the possibility to reset the output signal (Laplace-domain, Laplace-operator specified as **s**). The table shows the different forms to reset the output signal of the integration block.

Reset Signal	Trigger Set	Description
0		Summation of input signal.
>0	0	Assign the output signal permanently to the trigger value.
<0	0	Assign the output signal to the trigger value and continue summation of input signal with this value.
>0	1	Hold the last output value of output signal.
<0	1	Continued summation of input signal but one step delay.

The trigger signal is set once to '0' if the trigger event occurs; otherwise, the Integrator would be permanently reset.

$$y(k) = y(k-1) + KI \cdot TS \cdot x(k) \qquad \text{or} \qquad G(s) = \frac{KI}{s}$$

$$KI = \frac{1}{T_R} \qquad T_R = \text{Integral-Action Time} \qquad TS = \text{Sample Time}$$

Dialog Settings	
	«Integral Gain» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Initial Value» Defines the value for t=0. Enter a numerical value, a variable, or an expression in the text box to define the initial value at simulation start.
	«Upper Limit»/«Lower Limit» Define the value for the upper/lower limit of the output signal. Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The value '0' for upper and lower limit means there is no limitation.
☑ ☐	«Reset Integrator» If the box is checked, you can enter a reset signal, a reset value, and a control value in the corresponding text box.
    ☑ ☐	«Reset Signal» Controls the reset behavior of the integration block. There is a difference between =0 (no reset), from lower 0 to greater 0, and from greater 0 to lower 0. Usually a variable or a logical expression are chosen. Enter a numerical value, a variable, or an expression in the text box. «Use Pin» If the box is checked, the reset signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Reset Value» Defines the value after a reset event. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Trigger Setting» Defines the integration after a reset event. Each value <>0 means '1' in the table above. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼ ☑ ☐	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value. «Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
    ☑ ☐	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~). «Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Integral Gain	KI	real
Initial Value for t=0	Y0	real
Upper Limit	UL	real
Lower Limit	LL	real
Variable holding the Trigger Signal	TRIGSIG	real
Output Signal after Trigger Event	TRIGVAL	real
Controls the behavior after trigger event	CTRL	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Derivative

>>Basics>Blocks>Continuous Blocks

The block represents a differentiation block with limitation (Laplace-domain, Laplace-operator specified as **s**).

$$y(k) = \frac{KD}{TS} \cdot (x(k) - x(k-1)) \qquad \text{or} \qquad G(s) = KD \cdot s$$

$$KD = \text{Rate Time} \qquad TS = \text{Sample Time}$$

Dialog Settings	
	«Differential Gain» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Initial Value» Defines the value for t=0. Enter a numerical value, a variable, or an expression in the text box to define the initial value at simulation start.
	«Upper Limit»/«Lower Limit» Define the value for the upper/lower limit of the output signal. Enter a numerical value, a variable, or an expression in the text box. The value '0' for upper and lower limit means there is no limitation.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Differential Gain	KD	real
Initial Value for t=0	Y0	real
Upper Limit	UL	real
Lower Limit	LL	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Memory

>>Basics>Blocks>Continuous Blocks

The block represents a delay of one simulation step of the input signal with initial value. You cannot define a special sample time for the block.

$$y(k) = x(k-1)$$

Dialog Settings	
	«Initial Value» Defines the value for t=0. Enter a numerical value, a variable, or an expression in the text box to define the initial value at simulation start.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Initial Value for t=0	Y0[0]	real
Input Signal	INPUT	real

Component Outputs

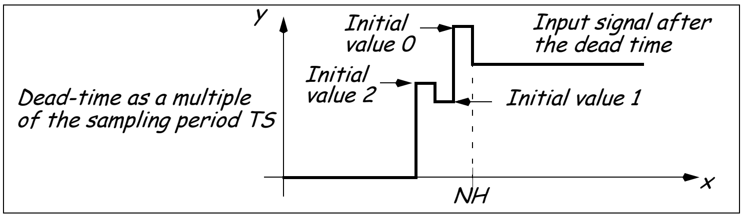
Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Dead-Time Element

>>Basics>Blocks>Continuous Blocks

The block represents a delay of a number of simulation steps of the input signal with initial values. The delay is specified as a number of samples.

$$
\begin{aligned}
y(k) &= x(k-NH) \quad \text{for} \quad k > NH \\
y(k) &= Y0(NH-k) \quad \text{for} \quad k \le NH \\
NH &= \text{Number of Samples}
\end{aligned}
$$



Dialog Settings	
	«Initial Values» Defines the initial values for k<NH. Click the **«Value»** field of the corresponding initial parameter. Enter a numerical value, a variable, or an expression to define the initial values at simulation start.
⊙	**«Separate»** The option defining single initial values is selected. You have to enter the initial values in the table on the right side. Create, delete, or move entries with the symbols placed on the upper right side. If you click the index name, you can edit the number (non-negative integer). Please note: You cannot define a index twice. See also “Defining Parameters” on page 49.
⊙	**«File»** The option using a file for initial values is selected. The initial values are defined by a data table. You can load an existing file and can modify the data when necessary in the lookup table. The number of samples is defined by the number of values in the file. See also “The .mdx File Format” on page 467.
	«Samples» Defines the number of cycles (samples) for which a signal should be delayed at simulation start. Enter a numerical value, a variable, or an expression. Please note: The value must be a positive integer. The value is not available for files.
▼	**«Sample Time»** Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	**«Use Pin»** If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Number of Samples (for separate initial values)	NH	real
Initial Values (for separate initial values)	Y0[n]	real
File with Initial Values (for initial values defined in a file)	FILE	file
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Delay

>>Basics>Blocks>Continuous Blocks

The block represents a delay time of the input signal with a defined number of samples. The delay is specified as simulation time.

$$y(t) = x(t-TD) \qquad TD = NH \cdot [t(k-1)-t(k)]$$
$$TD = \text{Delay Time} \qquad NH = \text{Samples}$$

Dialog Settings	
	«Delay Time» Defines the delay time in seconds. Enter a numerical value (non-negative integer with or without unit suffix) in the text box to define the delay time. Please note: You cannot define a variable or expression in this case. Only a number is accepted.
	«Samples» Defines the number of samples within the delay time. Enter a numerical value (non-negative integer with or without unit suffix) in the text box to define the samples. Please note: You cannot define a variable or expression in this case. Only a number is accepted.
	<Initial Values> Defines the values for t<TD. Open the dialog. Click the **«Value»** field of the corresponding initial value. Enter a numerical value, a variable, or an expression to define the initial values at simulation start.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).

☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Initial Values	Q0[n]	real
Input Signal	INPUT	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

5.2 Discrete Blocks

- Z-Transfer Function (GZ)
- Integrator (GZ)
- Sample & Hold (SAH)
- Unit Delay (GZ)
- Filter (GZ)

Z-Transfer Function (Discrete)

>>Basics>Blocks>Discrete Blocks

The Block represents time discrete transfer characteristic G*(z) by a difference equation. The block can be applied to model discontinuous controllers and processes. Within the dialog you can define numerator and denominator order and coefficients of the Z-Transfer Function. In almost all cases of analyzing dynamic processes in controller systems, vanishing initial values (identical to zero) can be assumed. If this assumption is not fulfilled, initial values can be assigned to the variables *x* and *y* according to the implementation structure of $G(z^{-1})$.

$$y(k) = q_0 x(k) + q_1 x(k-1) + \ldots + q_n x(k-n) - p_1 y(k-1) - p_2 y(k-2) - \ldots - p_m y(k-m)$$

$$\text{or} \qquad G(z^{-1}) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2} + \ldots + q_n z^{-n}}{p_0 + p_1 z^{-1} + p_2 z^{-2} + \ldots + p_m z^{-m}}$$

Dialog Settings

«Numerator»«Order»
Enter a numerical value (non-negative integer with or without unit suffix) in the text box to define the numerator order of the z-transfer function. Please note: You cannot define a variable or expression in this case. Only a number is accepted.

«Numerator»«Coefficient»
Click the «Value» field of the corresponding numerator coefficient Q[n] to define a value. Common parameter types. Enter a numerical value, a variable, or an expression.

«Denominator»«Order»
Enter a numerical value (non-negative integer with or without unit suffix) in the text box to define the denominator order of the z-transfer function. Please note: You cannot define a variable or expression in this case. Only a number is accepted.

«Denominator»«Coefficient»
Click the **«Value»** field of the corresponding denominaotr coefficient P[n] to define a value. Common parameter types. Enter a numerical value, a variable, or an expression.

<Initial Values>
Defines the initial values for X (Q[n]) and Y (P[n]). Enter a numerical value, a variable, or an expression in the text box to define the initial values at simulation start.

<From G(s)>
Transfers G(s) coefficients into G(z). Define the G(s) coefficients A[n] and B[n] and click <OK>. The G(s) functions are transferred into G(z).

$G(z)=\mathcal{Z}\{\mathcal{L}^{-1}[G(s)]\}$ with $\mathcal{L}^{-1}[G(s)]=g(t)$ and $t=k*TS$; general: $z=e^{s*TS}$

▾ «Sample Time»
Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.

☑ ☐ «Use Pin»
If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

«Input Signal»
Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).

☑ ☐ «Use Pin»
If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Dimension of Numerator	N	real
Dimension of Denominator	D	real
Numerator Coefficients	Q[n]	real
Denominator Coefficients	P[n]	real
Initial Values of X-Values	Q0[n]	real
Initial Values of Y-Values	P0[n]	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Discrete Integrator

>>Basics>Blocks>Discrete Blocks

The block represents an integration block with limitation and the possibility to reset the output signal (Laplace-domain, Laplace-operator specified as **s**). The table shows the different forms to reset the output signal of the integration block.

Reset Signal	Trigger Set	Description
0		Summation of input signal.
>0	0	Assign the output signal permanently to the trigger value.
<0	0	Assign the output signal to the trigger value and continue summation of input signal with this value.
>0	1	Hold the last output value of output signal.
<0	1	Continued summation of input signal but one step delay.

The trigger signal is set once to '0' if the trigger event occurs; otherwise, the Integrator would be permanently reset.

$$y(k) = y(k-1) + KI \cdot TS \cdot x(k) \qquad \text{or} \qquad G(s) = \frac{KI}{s}$$

$$KI = \frac{1}{T_R} \qquad T_R = \text{Integral-Action Time} \qquad TS = \text{Sample Time}$$

Dialog Settings	
	«Integral Gain» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Initial Value» Defines the value for t=0. Enter a numerical value, a variable, or an expression in the text box to define the initial value at simulation start.
	«Upper Limit»/«Lower Limit» Define the value for the upper/lower limit of the output signal. Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The value '0' for upper and lower limit means there is no limitation.
☑ ☐	«Reset Integrator» If the box is checked, you can enter a reset signal, reset value, and a control value in the corresponding text box.
	«Reset Signal» Controls the reset behavior of the integration block. There is a difference between =0 (no reset), from lower 0 to greater 0, and from greater 0 to lower 0. Usually a variable or a logical expression are chosen. Enter a numerical value, a variable, or an expression in the text box. See also the table above.
☑ ☐	«Use Pin» If the box is checked, the reset signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

	«Reset Value» Defines the value after a reset event. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Trigger Setting» Defines the kind of integration after a reset event. Each value <>0 means '1' in the table above. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Sample Time» Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Integral Gain	KI	real
Initial Value for t=0	Y0	real
Upper Limit	UL	real
Lower Limit	LL	real
Variable holding the Trigger Signal	TRIGSIG	real
Output Signal after Trigger Event	TRIGVAL	real
Controls the behavior after trigger event	CTRL	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

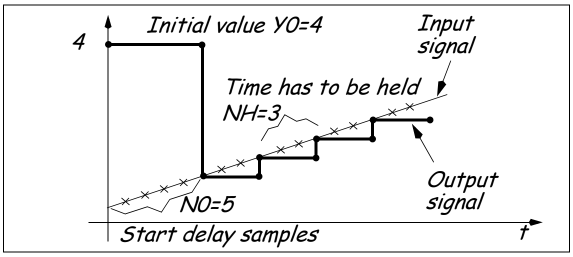
Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Sample and Hold Element

>>Basics>Blocks>Discrete Blocks

The block samples the input signal and holds this value constant up to the next sampling moment (discretization by time). The sampling period is specified as a number of samples.

$$y(k) = x(n) \qquad n = \mathrm{INT}\left(\frac{k - N0}{NH}\right) + N0 \qquad k > N0$$



Dialog Settings	
	«Initial Value» Defines the initial value of the input signal for N<N0. Enter a numerical value, a variable, or an expression in the text box to define the initial value at simulation start.
	«Start Delay Samples» Defines the number of samples for which the initial value should be held at simulation start. Enter a numerical value, a variable, or an expression in the text box. Please note: The value must be a positive integer.
	«Samples» Defines the number of samples for which a signal should be held. Enter a numerical value, a variable, or an expression in the text box. Please note: The value must be a positive integer.
▼	«Sample Time» Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Number of Samples	NH	real
Initial Samples	N0	real
Initial Value	Y0	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Unit Delay

>>Basics>Blocks>Discrete Blocks

The block represents a delay of the of the input signal with the as sample time defined time. You can define an initial value for the output signal.

$$y(k) = x(k-1)$$
$$TS = Value$$

Dialog Settings	
	«Initial Value» Defines the value for t<TS. Enter a numerical value, a variable, or an expression in the text box to define the initial value at simulation start.
▼	«Sample Time» Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Initial Value for t=0	Q0[1]	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Digital Filter

>>Basics>Blocks>Discrete Blocks

The block provides several filter types for the block diagram module. The filter models are implemented as a transfer function G(z) which is parametrized with parameter sets in the property dialog. See also Schrüfer E.: Signalverarbeitung, C. Hanser, Verlag, 1992.

Dialog Settings	
▼	«Filter» In the list, select the Butterworth, Bessel, Tschebyscheff, or Cauer filter.

▼	«Filter Type» In the list, select Low Pass, High Pass, Band Pass, or Band-Stop type.
	«Order» Defines the filter order. Enter a numerical value (non-negative integer with or without unit suffix) in the text box. Please note: You cannot define a variable or expression in this case. Only a number is accepted.
	«Cut-Off Frequency» Defines the cut-off frequency (Band Pass type has a lower and upper). Enter a numerical value (non-negative integer with or without unit suffix) in the text box. Please note: You cannot define a variable or expression in this case. Only a number is accepted.
▼ ☑ ☐	«Sample Time» Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value. «Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
  ☑ ☐	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~). «Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Dimension of Numerator	N	real
Dimension of Denominator	D	real
Numerator Coefficients	Q[n]	real
Denominator Coefficients	P[n]	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

5.3 Source Blocks

- Constant Value (CONST)
- Step Function (STEP)
- Random (RANDOM)

Constant Value

>>Basics>Blocks>Source Blocks

The block provides a value calculated at each simulation step.

`y(k) = K`

Dialog Settings	
	«Value» Defines the output value. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use Pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
AC Parameters	
⊙	«Phase & Magnitude» The Phase/Magnitude representation is selected. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also "AC Simulation Models" on page 401.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension of the phase.
⊙	«Real & Imaginary» The Real/Imaginary representation is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Constant Value	CONST	real
Sample Time [s]	TS	real

Component Outputs

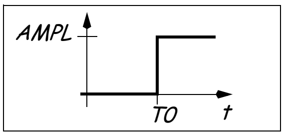
Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real
Magnitude AC Analysis	AC_MAG	real
Phase Shift AC Analysis [deg]	AC_PHASE	real

Step Function

>>Basics>Blocks>Source Blocks

The block provides a signal which changes at time t=T0 instantly from zero to the specified amplitude K.

$$
\begin{aligned}
y(k) &= 0 && \text{for } t < T0 \\
y(k) &= AMPL && \text{for } t \geq T0
\end{aligned}
$$



Dialog Settings	
	«Step time» Defines the step time T0. Enter a numerical value, a variable, or an expression in the text box. A value lower than '0' is considered as '0'.
	«Amplitude» Defines the amplitude. Enter a numerical value, a variable, or an expression in the text box.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
AC Parameters	
⊙	«Phase & Magnitude» The Phase/Magnitude representation is selected. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also “AC Simulation Models” on page 401.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension of the phase.
⊙	«Real & Imaginary» The Real/Imaginary representation is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Step Time	T0	real

Amplitude	AMPL	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real
Magnitude AC Analysis	AC_MAG	real
Phase Shift AC Analysis [deg]	AC_PHASE	real

Random

>>Basics>Blocks>Source Blocks

The block provides to each simulation step a random number within the defined range of values at the block output.

$$\mathrm{MIN} \leq y(k) \leq \mathrm{MAX}$$

Dialog Settings	
	«Maximum Value»«Minimum Value» Defines the maximum and minimum output value of the random function. Enter a numerical value, a variable, or an expression in the text box.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
AC Parameters	
⊙	«Phase & Magnitude» The Phase/Magnitude representation is selected. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. See also "AC Simulation Models" on page 401.
▼	«Angular Dimension» In the list, select radians or degrees for the angular dimension of the phase.
⊙	«Real & Imaginary» The Real/Imaginary representation is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Maximum Random Value	MAX	real
Minimum Random Value	MIN	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real
Magnitude AC Analysis	AC_MAG	real
Phase Shift AC Analysis [deg]	AC_PHASE	real

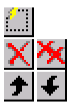
5.4 Signal Processing

- Maximum of Input Signals (MAX)
- Minimum of Input Signals (MIN)
- Summation (SUM)
- Multiplier (MUL)
- Negator (NEG)
- Limiter (LIMIT)
- Comparator (COMP)
- Two-Point Element with Hysteresis (TPH)
- N-Point Element (NP)
- Discretization Element (DIS)
- Nonlinear Characteristic (NL)
- Equation Block (FML)
- Maximum/Minimum Value at Time T (MAXT/MINT)
- Multidimensional Table Block (NDNL)

Maximum of Input Signals

>>Basics>Blocks>Signal Processing
The block provides the maximum value from all input signals at each time step at the block output. The component cannot be used with AC and DC simulation.

$$y(k) = \mathrm{Max}(x_1(k), x_2(k), \ldots, x_n(k))$$

Dialog Settings	
	**«Input Signals»** Create, delete, or move entries of input signals with the symbols placed on the upper right side. Please note: If you need more than the available places for pins at the symbol, clear the **«Use Pin»** box and type the name in the **«Input Signal»** text box. See also “Defining Parameters” on page 49.
	«Name» Displays the name of the input signals. If you click the index name, you can modify the number (non-negative integer). Please note: You cannot define a index twice.
	«Info» User specific field to enter information about an input signal. Click the field of the corresponding input signal and enter the text. You can display these information on the sheet by defining the display setting within the block output dialog. See also “Defining Displays and Outputs” on page 51.

▼	«Sign» In the list, select **«Plus»** or **«Minus»**. The sign is applied to the corresponding input signal.
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT[n]	real
Sample Time [s]	TS	real

Component Outputs

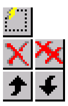
Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Minimum of Input Signals

>>Basics>Blocks>Signal Processing

The block provides the minimum value from all input signals at each time step at the block output. The component cannot be used with AC and DC simulation.

$$y(k) = \mathrm{Min}(x_1(k), x_2(k), \ldots, x_n(k))$$

Dialog Settings	
	«Input Signals» Create, delete, or move entries of input signals with the symbols placed on the upper right side. Please note: If you need more than the available places for pins at the symbol, clear the **«Use Pin»** box and type the name in the **«Input Signal»** text box. See also “Defining Parameters” on page 49.
	«Name» Displays the name of input signals. If you click the index name, you can modify the number (non-negative integer). Please note: You cannot define a index twice.
	«Info» User specific field to enter information about an input signal. Click the field of the corresponding input signal and enter the text. You can display these information on the sheet by defining the display setting within the block output dialog. See also “Defining Displays and Outputs” on page 51.
▼	«Sign» In the list, select **«Plus»** or **«Minus»**. The sign is applied to the corresponding input signal.

☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT[n]	real
Sample Time [s]	TS	real

Component Outputs

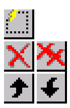
Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Summation

>>Basics>Blocks>Signal Processing

The block adds up all input signals at each time step and provides the result at the block output.

$$y(k) = x_1(k) + x_2(k) + \ldots + x_n(k)$$

Dialog Settings	
	«Input Signals» Create, delete, or move entries of input signals with the symbols placed on the upper right side. Please note: If you need more than the available places for pins at the symbol, clear the **«Use Pin»** box and type the name in the **«Input Signal»** text box. See also “Defining Parameters” on page 49.
	«Name» Displays the name of the input signals. If you click the index name, you can modify the number (non-negative integer). Please note: You cannot define a index twice.
	«Info» User specific field to enter information about an input signal. Click the field of the corresponding input signal and enter the text. You can display these information on the sheet by defining the display setting within the block output dialog. See also “Defining Displays and Outputs” on page 51.
▼	«Sign» In the list, select **«Plus»** or **«Minus»**. The sign is applied to the corresponding input signal.

☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT[n]	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Multiplier

>>Basics>Blocks>Signal Processing

The block multiplies all input signals at each time step and provides the result at the block output.

$$y(k) = x_1(k) \cdot x_2(k) \cdot \ldots \cdot x_n(k)$$

Dialog Settings	
	«Input Signals» Create, delete, or move the entries of input signals with the symbols placed on the upper right side. Please note: If you need more than the available places for pins at the symbol, clear the **«Use Pin»** box and type the name in the **«Input Signal»** text box. See also “Defining Parameters” on page 49.
	«Name» Displays the name of the input signals. If you click the index name, you can modify the number (non-negative integer). Please note: You cannot define a index twice.
	«Info» User specific field to enter information about an input signal. Click the field of the corresponding input signal and enter the text. You can display these information on the sheet by defining the display setting within the block output dialog. See also “Defining Displays and Outputs” on page 51.
▼	«Sign» In the list, select **«Plus»** or **«Minus»**. The sign is applied to the corresponding input signal.

☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT[n]	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Negator

>>Basics>Blocks>Signal Processing

The block multiplies the input signal at each time step with -1 and provides the result at the block output.

$$y(k) = -x(k)$$

Dialog Settings	
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Limiter

>>Basics>Blocks>Signal Processing

The block represents a limitation of the input signal within defined limits.

$$\begin{aligned} y(k) &= \text{Lower Limit. if } x(k) \le \text{Lower Limit} \\ y(k) &= \text{Upper Limit, if } x(k) \ge \text{Upper Limit} \\ y(k) &= x(k) \text{ else} \end{aligned}$$

Dialog Settings	
	«Upper Limit»/«Lower Limit» Define the value for the upper/lower limit of the output signal. Common parameter types. Enter a numerical value, a variable, or an expression in the text box. The value '0' for upper and lower limit means there is no limitation.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Upper Limit of Output Signal	UL	real
Lower Limit of Output Signal	LL	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

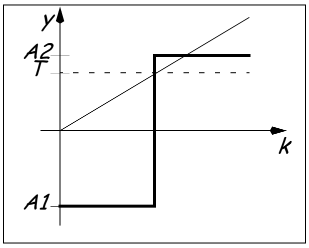
Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Comparator

>>Basics>Blocks>Signal Processing

The block represents a two-point element **without** hysteresis.

$$y(k) = A1, \text{ if } x(k) < T$$
$$y(k) = A2, \text{ if } x(k) \geq T$$



Dialog Settings	
	«Threshold» Defines the threshold of the comparator. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Value A1»«Value A2» Define the amplitude left (A1) and right (A2) from the threshold. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Threshold	T	real
Value A1	VAL1	real
Value A2	VAL2	real

Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

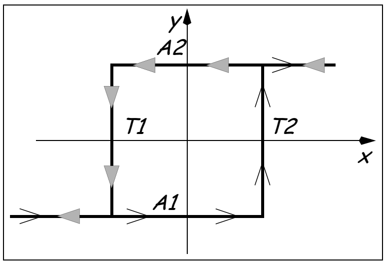
Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Two-point Element with Hysteresis

>>Basics>Blocks>Signal Processing

The block represents a two-point element **with** hysteresis. The component cannot be used with AC and DC simulation.

$$\begin{aligned}
y(k) &= A1, \text{ if } x(k) < T1 \\
y(k) &= A1, \text{ if } x(k) < T2 \text{ and } A(k-1) = A1 \\
y(k) &= A2, \text{ if } x(k) > T2 \\
y(k) &= A2, \text{ if } x(k) > T1 \text{ and } A(k-1) = A2
\end{aligned}$$



Dialog Settings	
	«Threshold 1»«Threshold 2» Define the threshold T1 and T2 of the Two-Point Element. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
	«Value A1»«Value A2» Define the amplitude left (A1) from the threshold T1 and right (A2) from the threshold T2. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.
▼	**«Sample Time»** Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	**«Use Pin»** If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	**«Use Pin»** If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Threshold T1	T1	real
Threshold T2	T2	real
Value A1	VAL1	real
Value A2	VAL2	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

N-Point Element

>>Basics>Blocks>Signal Processing

The input signal x(k) is compared with n thresholds starting from the lowest. If the value of x is greater than to the n^{th} threshold the (n+1) threshold amplitude is given to the output. If the input signal is lower than the smallest n^{th} threshold, the n^{th} threshold amplitude is issued._

The user must always define one amplitude more than the number of thresholds.

$$\begin{array}{lll} \text{If } x(k) > THRES(n) & y(k) = AMPL(n+1) & \\ \text{If } x(k) \leq THRES(n) & y(k) = AMPL(n) & THRES(n) \rightarrow \text{smallest threshold} \end{array}$$

Dialog Settings	
	«Number of Thresholds» Defines the threshold number. Enter a numerical value (non-negative integer with or without unit suffix) in the text box. Please note: You cannot define a variable or expression in this case. Only a number is accepted.
	«Threshold» Defines the threshold values. Common parameter types. Click the **«Value»** field of the corresponding threshold THRES[n] to define a value. Common parameter types. Enter a numerical value, a variable, or an expression.
	«Value» Defines the corresponding output amplitude of a threshold. Common parameter types. Click the **«Value»** field of the corresponding threshold amplitude AMPL[n] to define a value. Common parameter types. Enter a numerical value, a variable, or an expression.
▼	**«Sample Time»** Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value. **«Use Pin»** If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
☑ ☐	

	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Number of Thresholds	NTHRES	real
Threshold	THRES[n]	real
Value	AMPL[n]	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

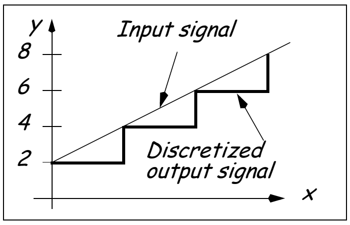
Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Discretization Element

>>Basics>Blocks>Signal Processing

The input signal is discretized with respect to the specified coefficient (quantification of the signal amplitude). In other words, the minimum signal increment or decrement is always equal to K.

$$y(k) = K \cdot INT\left(\frac{x(k)}{K}\right)$$



Dialog Settings	
	«Discretization Quantity» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▾	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Discretization Quantity	K	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Nonlinear Transfer Function

>>Basics>Blocks>Signal Processing

The block represents a nonlinear transfer behavior.

$$y(k) = f(x(k))$$

Dialog Settings

«Characteristic»
To describe the nonlinear relation, you must specify a name of an appropriate XY nonlinear characteristic function. See also “Separate Component Characteristic” on page 297.

To define characteristic data you can do one of the following:

- **«Use Pin»**; the XY component, connected to the pin, provides the characteristic.

Within the Characteristic dialog:

- **«Reference»**; the name in the text box specifies an XY element placed on the sheet.
- **«Lookup Table»**

a. **«File Reference»**; the name in the text box specifies a variable which refers to a subsheet port specifying a characteristic file.

b. **«Lookup Table»**; data pairs, defined in the table, are used for characteristic.

- ☑ **«File Name»;** if you use an file, data pairs are displayed in the lookup table and used for characteristic. You can modify the data when necessary.

▼ «Sample Time»
Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.

☑ ☐ «Use Pin»
If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

«Input Signal»
Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).

☑ ☐ «Use Pin»
If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Characteristic	CH	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Equation Block

>>Basics>Blocks>Signal Processing
The block provides a numerical links between input signals and block intern variables at the block output.

$$y(m) = f(x(n), x(n-1), \ldots, var1, var2, \ldots)$$
$$y(m) = VAL(m) \qquad x(n) = INPUT(n)$$

Dialog Settings

«Name»
Displays the number of the input signals. If you click the index name, you can modify the number (non-negative integer). Please note: You cannot define a index twice.

«Input Signals»
Create, delete, or move the entries of input signals with the symbols placed on the upper right side. Please note: If you need more than the available places for pins at the symbol, clear the **«Use Pin»** box and type the name in the **«Input Signal»** text box. See also “Defining Parameters” on page 49.

☑ ☐ «Use Pin»
If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

«Type»
Create, delete, or move entries of block intern variables or output signals with the symbols placed on the upper right side. Please note: The variables defined within the block cannot be used for other components. See also “Defining Parameters” on page 49.

	«Name» Defines the names for block intern variables. Enter the name for the internal variable.
	«Equation» Defines the numerical link between the input and output signal. You can use all variables defined within the equation block, mathematical standard functions and names of the block input signals, e.g. `INPUT[0]`. Please note: You cannot use variables outside from the equation block.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal [n]	INPUT[n]	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Maximum/Minimum Value at Time T

>>Basics>Blocks>Signal Processing

The block provides the maximum/minimum input signal to the current simulation time at the block output. The defined initial value applies at the block output as long as the input signal has crossed this value at least once. The component cannot be used with AC and DC simulation.

$$y(k) = \mathrm{Max}(x(k)) \cdot \mathrm{FACT}, \text{ if } \quad x(k) > \mathrm{Max}(x(k-1)) \wedge x(k) > \text{Initial value}$$
$$k = \sum h, \quad h \text{ Simulation step}$$

$$y(k) = \mathrm{Min}(x(k)) \cdot \mathrm{FACT}, \text{ if } \quad x(k) < \mathrm{Min}(x(k-1)) \wedge x(k) < \text{Initial value}$$
$$k = \sum h, \quad h \text{ Simulation step}$$

Dialog Settings	
▼	«Function» In the list, select **«MaximumT»** for the maximum or **«MinimumT»** for the minimum function.
	«Initial Value» Defines the value which applies at the block output as long as the input signal has crossed the initial value at least once. Enter a numerical value, a variable, or an expression in the text box to define the initial value at simulation start

	«Factor» Defines the factor which is multiplied by the output signal. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT	real
Adaption factor for the output signal	FACT	real
Sample Time	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

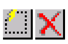
Multidimensional Table Block

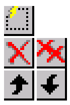
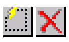
>>Basics>Blocks>Signal Processing

The multidimensional table block provides a table of output signals which dependent on different, nonlinear input signals. Each input signal (non-conservative node) has a specific characteristic with extrapolation type. The characteristics of input signals can also be imported from a SIMPLORER .mdx file created in the DAY Post Processor.

```
val[n](t) = f(x(QUANT)[m])
```

Dialog Settings	
	«Name» Displays the component internal names of the input quantities which are defined in the <Create/Edit Table> dialog. You cannot make changes in these fields.
	«Input» The values of the defined signal control the values of output signals. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect the quantity.

☑ ☐	«Use pin» If the box is checked, the signal is defined by the quantity connected to the pin, otherwise the name in the text box is used.
▼	«Extrapolation» Click the list field to select the **«Linear»**, **«Periodic»**, **«Halfperiod»**, **«Constant»**, **«Even»**, or **«Odd»** extrapolation type to define the quantity characteristics outside the range of parameter sweep. The default extrapolation is **«Linear»**. Select **«Linear»** to extrapolate a linear waveform. This option takes the last two points of the wave and generates a straight line extending beyond the range of parametric sweep. Select **«Periodic»** to repeat the wave outside the range of parametric sweep. Select **«Halfperiod»** to mirror then repeat the waveform outside the range of parameter sweep. Select **«Constant»** to extrapolate a constant value from the last point in the interpolation. Select **«Even»** to repeat the wave outside the range of parametric sweep. Select **«Odd»** to repeat a reflection of the waveform outside the range of the parametric sweep. For example, extrapolating with this option reflects the wave into the positive range.
▼	«Interpolation» Click the list field to select the «NOSPLINE», «LASTSPLINE», or «DEEPSPLINE» interpolation type to define the quantity characteristic inside the range of parameter sweep. The default interpolation is «NOSPLINE». The interpolation type is valid for all input quantities. Select «NOSPLINE» to fit a straight line between the points of interpolation. Select «LASTSPLINE» to apply a Bezier interpolation to the last quantity characteristic. Select «DEEPSPLINE» to apply a Bezier interpolation to all quantity characteristics.
☑ ☐	«File Name» If the box is checked, characteristics of input values are defined by the file entered in the text box, otherwise the values in the <Create/Edit Table> dialog are used. Please note: Select the correct file filter (.mdx) before browsing.
	<Create/Edit Table> Click to open the dialog to create or edit input and output signals.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
Input Quantities	
	«Edit Input Channels» Create or delete entries with the symbols placed on the upper right side.
	«Index» Displays the component internal index of the input signals. You cannot make changes in these fields.
	«Name»«Unit» Click in the fields to define names and units for the corresponding input signal. Both are displayed in the column headings of the lookup table and saved in the created .mdx file.

	«Channel Data» Select an input signal in the list to define the corresponding characteristic data in the lookup table. The name of the active input signal displays the group box inscription. Create, delete, or move entries with the symbols placed on the upper right side. Click in the fields to enter values. Please note, you cannot use variables or expressions. Only numbers are accepted.
☑ ☐	«Sweep Data» If the box is checked, you can sweep data for a given range of values. <Set> replaces the values in the lookup table, <Append> inserts the new data at the table end.
Output Quantities	
	«Edit Output Channels» Create or delete entries with the symbols placed on the upper right side. Each new output signal represents a non-conservative node.
	«Index» Displays the component internal index of the input quantities. You cannot make changes in these fields.
	«Name»«Unit» Click in the fields to define names and units for the corresponding output signal. Both are displayed at the top of the columns in the lookup table and saved in the created .mdx file.
	«Edit Output Data» Click in the corresponding field to enter values for output signal. You can also use the <Tree> mode. Both views are automatically updated. Please note, you cannot use variables or expressions. Only numbers are accepted.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT[n]	real
Table File Name	FILE	real
Interpolation Type	IP	real
Extrapolation Type	EP[n]	real
Sample Time [s}	TS	real

Simulator Outputs

Description [Unit]	Parameter Name	Data Type
Output Vector	VAL[n]	real
Derivative of Output Vector	dVAL[n][m]	real

5.5 Math

- Sine (FCT)
- Cosine (FCT)
- Tangent (FCT)
- Sine Hyperbolic (FCT)
- Arc Cosine (FCT)
- Exponential Function (FCT)
- Natural Logarithm (FCT)
- Absolute Value (FCT)
- Reciprocal Value (FCT)
- Power (POW)
- Root (ROOT)
- Sign (SIGN)

Analytical Functions

>>Basics>Blocks>Math Blocks

These blocks allow the determination of the transfer characteristic according to certain analytical functions. Sine, Cosine, Tangent, Exponential function, Sine hyperbolic, Arc cosine, Natural Logarithm, Absolute value, Reciprocal value 1/x.

$$y(k) = f(x(k))$$

Dialog Settings	
▼	«Angle Unit» (Only for trigonometric functions) In the list, select radians or degrees. The dimension is applied to the input signal.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Power

>>Basics>Blocks>Math Blocks

The block raises the block input signal to the n-th power. The Exponent n has to be a non-negative fractional number and must be entered in decimal style.

$$y(k) = x(k)^n$$

Dialog Settings	
	«Exponent» Defines the exponent (fractional number) applied to the input signal. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Exponent	EXP	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Root

>>Basics>Blocks>Math Blocks

The block computes the n-th radical of the block input signal and provides the result at the block output. The radical exponent must be a positive natural number.

$$y(k) = \sqrt[n]{x(k)}$$

Dialog Settings	
	«Radical Exponent» Defines the radical exponent (fractional number). Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	**«Sample Time»** Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	**«Use Pin»** If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	**«Use Pin»** If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Radicand	ROOT	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

Sign

>>Basics>Blocks>Math Blocks

The block provides the discretized signal at the block output, depending on the sign of the block input signal and the adaption factor.

$$\begin{aligned} y(k) &= FACT && \text{if} \quad x(k) \geq 0 \\ y(k) &= -FACT && \text{if} \quad x(k) < 0 \end{aligned}$$

Dialog Settings	
	«Value» Defines the absolute Value of the output amplitude. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Absolute Value of Output Amplitude	AMPL	real
Input Signal	INPUT	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Block output signal	VAL	real

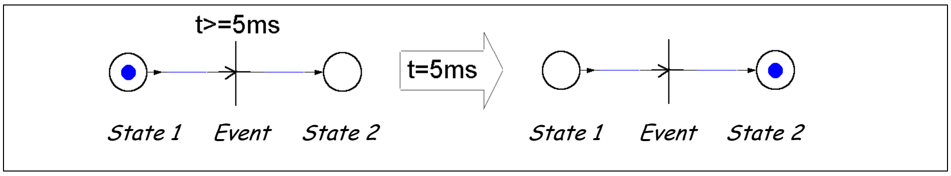
6 Modeling with State Graphs

The following components to create state graphs appear in the *States* folder of the **«Basics»** tab. Open the folder *States*, select a state or transition, drag the component onto the sheet and release the mouse button.

With SIMPLORER's state graph, discontinuous processes can be modeled as event-oriented based on the Petri Net theory. The theoretical basis of the modeling is to divide a system into:

- significant states and
- events, or transfers from one state to the other

A process sequence can be considered as a sequence of states. The current state is called active. Switching the activity from states to their successor states is called an event. An event occurs only if all previous states are active, all following states are inactive and the transfer condition in the form of a logical expression is true. At the beginning of the simulation, one state must be defined as active.



The event occurs if

- all previous states are active (marked) and
- the transfer conditions are TRUE; if no transfer condition is specified, the default value FALSE is used.

The result of an event will be

- all previous states are deactivated and
- all successor states are activated

On entering a state, several arithmetic and logical instructions (actions) that influence the system behavior can be executed.

You cannot use state graphs in AC and DC simulations. To leave state graphs on the sheet, select the components and choose ELEMENT>DON'T ADD TO MODEL DESCRIPTION. When you want to process a transient simulation, you can change this option.

Creating State Graphs

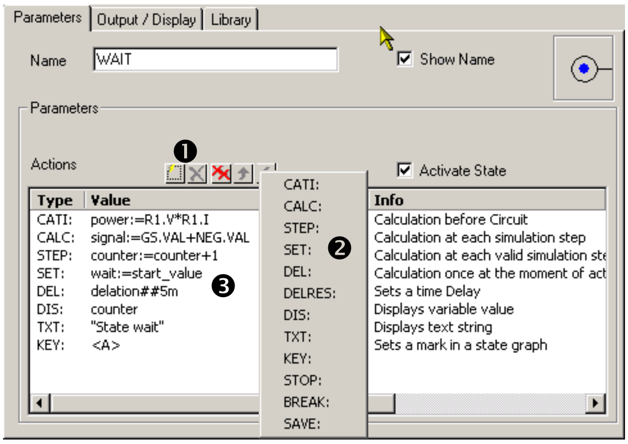
The state graph components are placed in the *States* folder in the **«Basics»** tab. Select the folder, click the component name (transition or state), drag the component onto the sheet and release the mouse button. Within the dialogs you can define the transfer conditions (transition) and actions (states). A selected component can be rotated (R key) or flipped (F key).

To connect the state graph components start the wire mode with CTRL+W or choose CONNECT>WIRE. Place the cross wires on the component pins and set beginnings, corners, and ends of the wire with the mouse. Exit the wire mode with ESC.

Schematic does not test the pins during the routing and allows all connections. However, only reasonable state graphs lead to correct simulation results.

Using Actions in State Graphs

For every state actions can be defined which are processed if the state is active. In the dialog for the State component, the available action types can be selected from a drop-down-list, which is opened by clicking on  .



❶ *Create a new action*
❷ *Select action type*
❸ *Enter the assignment*

<u>Actions in States</u>

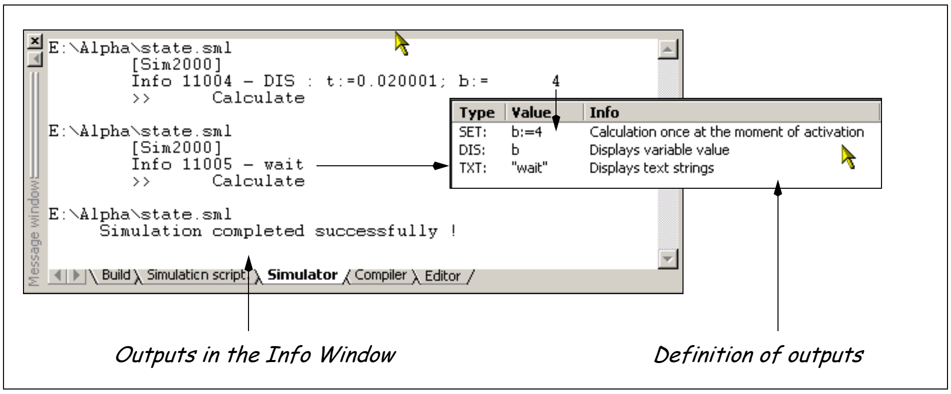
Action	Description	Syntax in Value
CALC	The variable is calculated at each simulation step and each transition from one state to another.	*var1:=100*t*
STEP	The variable is calculated at each **valid** simulation step.	*var2:=2*t*
CATI	The variable is calculated outside the state graph and before the calculation of the electric circuit.	*var3:=sqrt(t)*
SET	The variable is calculated only once at the moment of activation of the state.	*var4:=2.5*
DEL	Sets a delay. The variable is set to false at the moment of activation and set to true after the delay time.	var##time [s] *delation##10m*
DELRES	Deletes a defined delay variable.	del4
DIS	The variable value (and moment) is displayed in the simulator status window.	Name.Qualifier *dc.n*
TXT	The given text string is displayed in the simulator status window.	"Text String" "*State waiting*"
KEY	Sets a mark in the state graph by pressing a key.	*<A>*
STOP	Interrupts the simulation (can be continued).	No Parameters
BREAK	Finishes the simulation.	No Parameters
SAVE	Saves the active simulation status in a status file.	No Parameters

Differences in Assignment Actions in State Graphs

The action types CALC, STEP, and CATI differ in the sequence and computation frequency used by the simulator (that means, at each simulation step, at each valid step, outside the state graph). See also 13.6 “Simulator Backplane” on page 418.

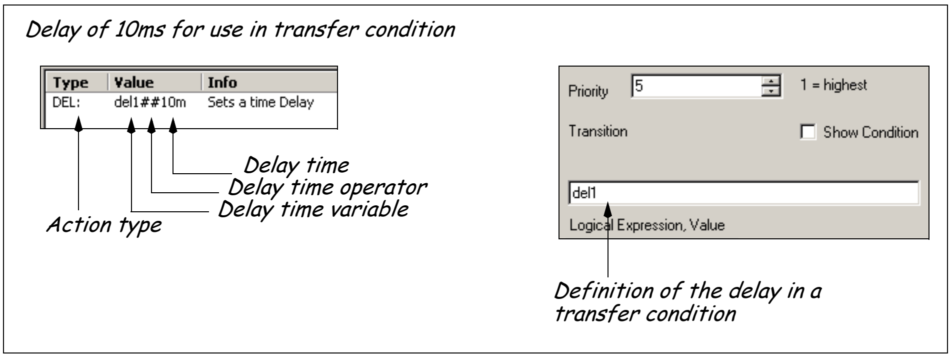
Displaying Variable Values and Text Strings by Action Types

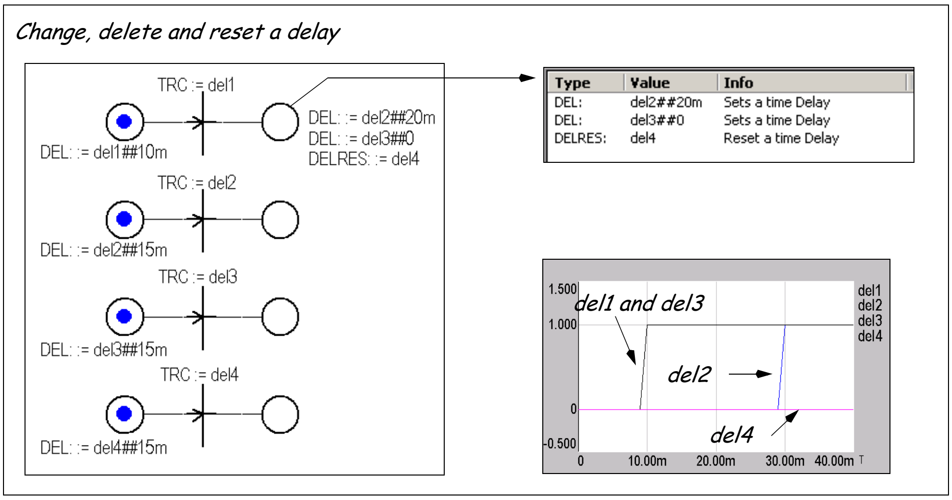
The action types DIS and TXT show the value of a variable (and moment) and a user-defined text string. The simulator writes these messages into the **«Simulator»** Information Window. To see the outputs, you must display this window. A user-defined text string must not contain a vowel mutation (umlaut). Please note: The text must be surrounded by quotation marks.



Defining and Resetting Delays

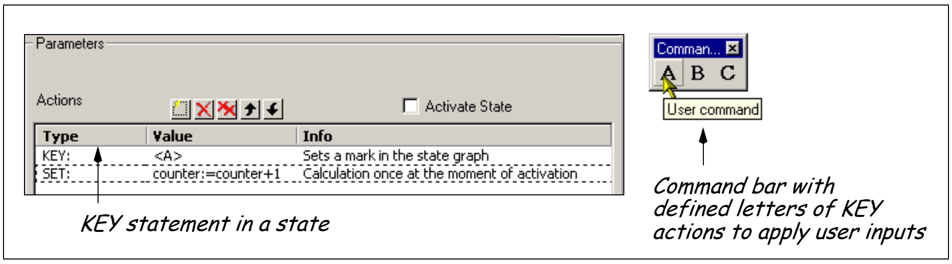
The action type DEL set a delay in seconds [s]. Throughout the stated delay time, the variable is set to FALSE, and when the delay time is over, set to TRUE. The variable in the DEL statement can be used in a transfer condition to set a delay within a state graph. The delay time can be reset, deleted or changed in other states of the simulation model.





Setting States with Key Actions

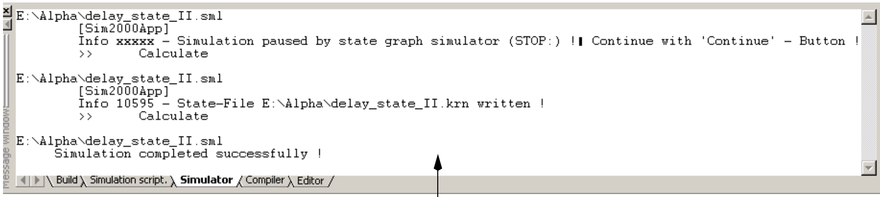
The KEY statement in a state enables the user to mark a state or states by clicking on the corresponding symbol in the KEY command bar **during the simulation**.



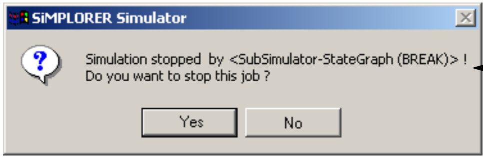
After the simulation starts, the KEY command bar is displayed. It contains a letter for each KEY statement. You can show or hide the command bar using the commands on the VIEW>TOOLBARS menu.

Stopping Simulations

The action types STOP, BRK, and SAVE influence the simulation process, in addition to the simulator functions. If a state is marked with one of these action types, the simulation is stopped (STOP) or paused (BRK). The SAVE statement automatically writes the active simulation state into the model file (.krn).



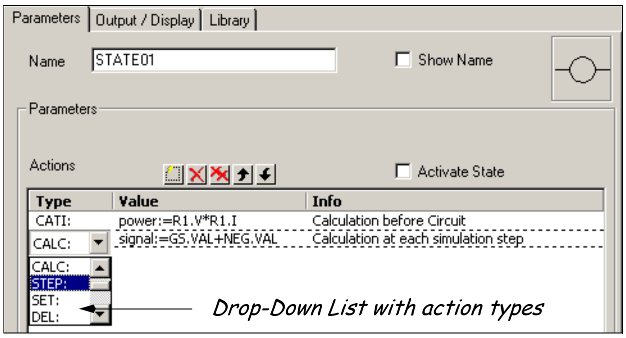
Messages in the Simulator Info Window using STOP or SAVE statement



Confirmation of BRK statement

Changing Action Types

You can change the action types of previously defined actions. You can also modify the already existing value assignment. Double-click the symbol or choose **«Properties»** in the shortcut menu to open the component dialog of a state. Click the name of action type you want to change and select a new statement in the drop-down list. Confirm your entries with <OK>.



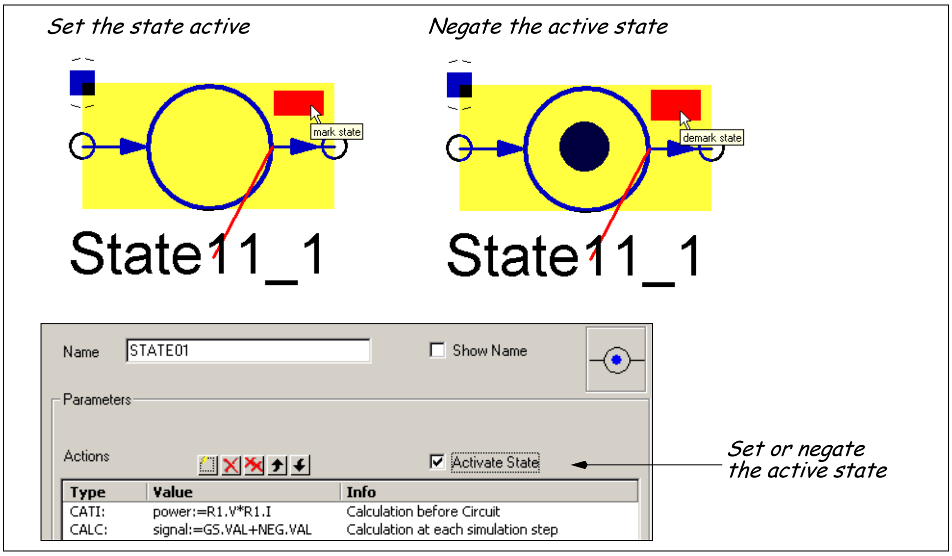
State Markers in State Graphs

Setting the Active State at the Beginning of the Simulation

The active state defines the beginning of the simulation in the state graph. Every state graph must have one active state at beginning of the simulation. A state can be set as active in the graphic display by clicking the red switch area or in the component dialog by selecting the **«Activate State»** option.

Selecting a switch changes the current state from active to inactive and vice versa. A blue circle indicates the state is active.

The switch area is visible only if the state element is selected.



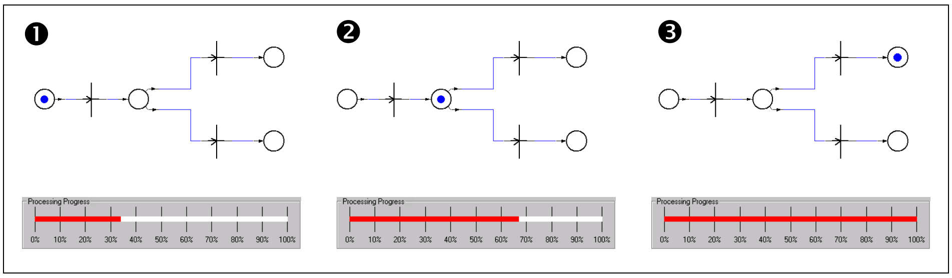
State Markers During a Simulation

During a simulation, the state markers are shown “running” through the state graph, depending on the active states in the simulation process.

If you want to see the state markers changing, select the option **«Use Animated Symbols»** in SHEET>PROPERTIES **«System»** and update the state graph elements from SIMPLORER version 4.2 with SHEET>SYNCHRONIZE.

Moving a State Marker

The SIMULATION>RESTORE ANIMATED SYMBOLS function initializes the state components if the simulation was interrupted.



Displaying the State of State Components

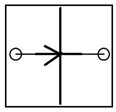
To display a state select either the output for Current value of State in the **«Output/Display»** tab or the parameter ST of a state component in a Display Element. The parameter ST has the direction attribute INOUT.

State Graph Components

- Transition Component
- States
- State Flexible

Transition Component

>>Basics>State Graph



With the transition component, the transfer condition between the states of the system can be described. The transition component is switched if all previous states are marked and the transfer condition is true. In case of an event, all predecessor states are deactivated and all successor states are activated. If in the logical expression of the transfer condition >= or <= used, the simulator is synchronized exactly on this event. The event is calculated with a precision of `Hmin`/2. If two events occur at the same time, that with the smaller priority number is computed first.In the dialog for the transition component, you define the transfer condition and the priority of the process, with '1' being the highest and '9' the lowest priority.

As part of the transfer condition any parameters from the simulation model and mathematical expressions can be used. If the equal operator is used, the synchronization is obtained with the minimum time step. This process can sometimes increase the simulation time. The default value for transfer conditions is FALSE.

Dialog Settings	
⬍	«Priority» Select the priority of the transfer condition; '1' is the highest and '9' the smallest priority.
	«Transition» Enter a numerical value, a variable, a mathematical or logical expression in the text box. The expression is interpreted as TRUE if the provided value is '1'. Otherwise the value is FALSE. See also “Equations, Expressions, and Variables” on page 32.

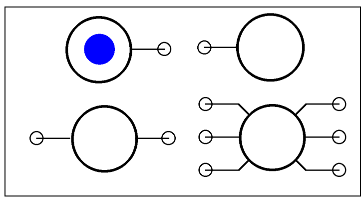
Component Parameters

Description [Unit]	Parameter Name	Data Type
Transfer Transition	TRC	real
Priority	PRI	real
Previous States	ST_PRE	real
Successor States	ST_SUC	real

State Components

>>Basics>State Graph

There are various symbols for states with different inputs and outputs. The choice of symbols depends on the structure of the state graph to be modeled.



Dialog Settings	
	«Type» Create, delete, or move entries for actions with the symbols placed on the upper right side. See also “Defining Parameters” on page 49.
	«Value» Type a numerical value (constant with or without unit suffix), a variable, a mathematical or logical expression in the text box. See also “Using Actions in State Graphs” on page 242.
	«Info» Field for user specific information about an expression. Click in the field of the corresponding action and enter the text. You can display these information on the sheet by defining the display setting within the **«Output/Displays»** tab.
☑ ☐	«Output» If the box is checked, the variable is saved in the result file (.sdb) during simulation. Please note: If you want to display a variable value, you have to select the box in the output dialog or in a display element. If the box is deactivated, you cannot display the variable value after the simulation. See also “Defining Displays and Outputs” on page 51.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Pin n	ST[n]	real
Define the Active State. Only at the beginning of a simulation.	ST0	real
Actions defined in the state. Please note: You cannot display these actions.	CATI: CALC: STEO: SET: DEL: DELRES: DIS: TXT: KEY: STOP: BREAK: SAVE:	

Component Outputs

Description [Unit]	Parameter Name	Data Type
Current Value of State	ST	real
Dynamic Value Output of a variable defined within the dialog of the state.	*Variable Name*	real

State Flexible

>>Basics>State Graph

The State Flexible Component represents a state with 30 pins. This is the maximum number of pins available for a state component. All pins should be connected with a transfer component. If there are open pins on the sheet, you can hide the pin by deactivating the pin **within the state output dialog**.

All other properties are identical with the state component described in “State Components” on page 247.

7 Using Measuring Instruments

SIMPLORER provides measuring instruments for different physical domains. These meters provide through and across quantities as well as power for every physical domain used in a simulation model. All meters are ideal components, that means, no relevant quantities are assumed to be '0'.

Except for the position sensors of the velocity-force representation, meters have no user-defined parameters. They have at least one conservative node and are connected with a specific physical domain circuit.

The meters can provide the measured value at a special output pin, which can be connected with input parameters in the Schematic.

Measuring instruments can be used for modeling in several ways.

- For physical domain components, measuring instruments can provide through, across, and power quantities for controlled sources and components.
- For blocks, measuring instruments can provide setpoints, disturbances and reference values.
- For state graphs, measuring instruments can provide values to define comparison functions and synchronization signals.

Meters for following domains are available:

- Electrical
- Fluidic
- Magnetic
- Mechanical
- Thermal

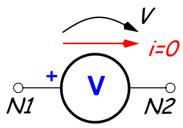
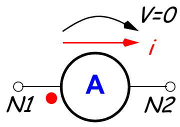
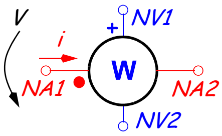
Measuring instruments appear in the *Measurement* folder of the **«Basics»** tab. Select the folder *Measurement*, open one of the physical domain folder, choose a meter for the corresponding quantity, drag the component onto the sheet and release the mouse button.

You cannot use fluidic, magnetic, mechanical, and thermal components in AC and DC simulations. To leave the components on the sheet, select the components and choose ELEMENT>DON'T ADD TO MODEL DESCRIPTION. When you want to process a transient simulation, you can change this option.

Electrical Meters

>>Basics>Measurement>Electrical

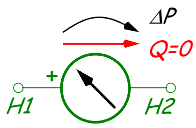
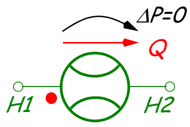
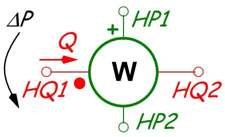
Electrical meters provide voltage (across quantity), current (through quantity), and power for electrical domain components.

Name/Equation	Symbol and Nodes	Outputs
Voltmeter (VM) $i(t) = 0 \quad R_i = \infty$		V [V]…Voltage dV [V/s]…Derivative of Voltage
Ammeter (AM) $\Delta v(t) = 0 \quad R_i = 0$		I[A]…Current dI [A/s]…Derivative of Current
Wattmeter (WM) $POWER(t) = v(t) \cdot i(t)$		P[W]…Power V [V]…Voltage I[A]…Current dV [V/s]…Derivative of Voltage dI [A/s]…Derivative of Current

Fluidic Meters

>>Basics>Measurement>Fluidic

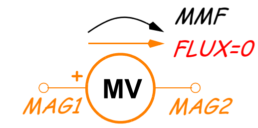
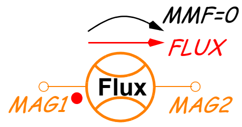
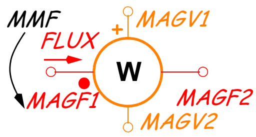
Fluidic meters provide pressure (across quantity), flow rate (through quantity), and power for fluidic domain components.

Name/Equation	Symbol and Nodes	Outputs
Manometer (PM) $Q(t) = 0$		P [Pa]…Differential Pressure
Flow Meter (QM) $\Delta P(t) = 0$		Q [m³/s]…Flow Rate
Wattmeter (WM_HYD) $POWER(t) = Q(t) \cdot P(t)$		P [Pa]…Differential Pressure Q [m³/s]…Flow Rate POWER [W]…Power

Magnetic Meters

>>Basics>Measurement>Magnetic

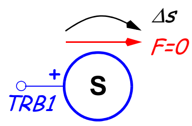
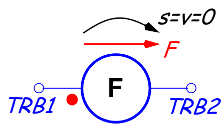
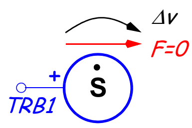
Magnetic meters provide magnetic voltage (across quantity), magnetic flux (through quantity), and power for magnetic domain components.

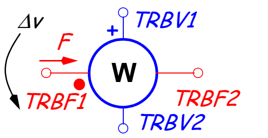
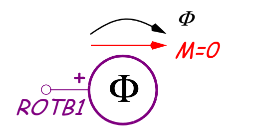
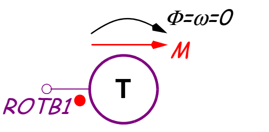
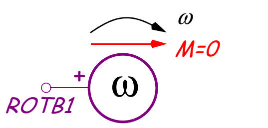
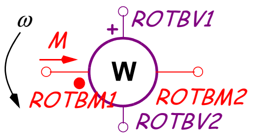
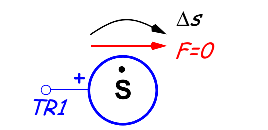
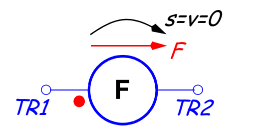
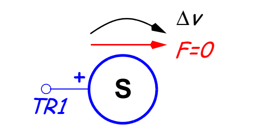
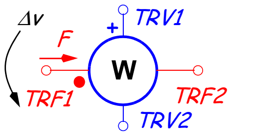
Name/Equation	Symbol and Nodes	Outputs
Magnetic Voltmeter (VM_MAG) $\mathrm{FLUX}(t) = 0$		MMF [A]...Magnetic Voltage
Flux Sensor (FLUXM) $\Delta\mathrm{MMF}(t) = 0$		FLUX [Vs]...Magnetic Flux
Wattmeter (WM_MAG) $\mathrm{POWER}(t) = \mathrm{FLUX}(t) \cdot \mathrm{MMF'}(t)$		MMF [A]...Magnetic Voltage FLUX [Vs]...Magnetic Flux POWER [W]...Power

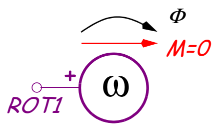
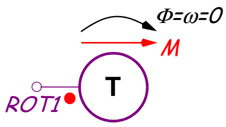
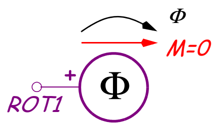
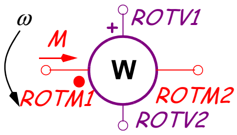
Mechanical Meters

>>Basics>Measurement>Mechanical

Mechanical meters provide position (across quantity), velocity (across quantity), force (through quantity), and power for mechanical domain components. There are meters for Displacement-Force and Velocity-Force representation. Each representation form has components for translational and rotational systems.

Name/Equation	Symbol and Nodes	Inputs and Outputs
Displacement-Force-Representation—Translational		
Position Sensor (SM_TRB) $F(t) = 0$		S [m]...Displacement
Force Sensor (FM_TRB) $\Delta s(t) = 0 \qquad \Delta v(t) = 0$		F [N]...Force
Velocity Sensor (VM_TRB) $F(t) = 0$		V [m/s]...Velocity

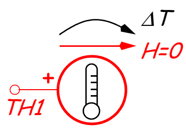
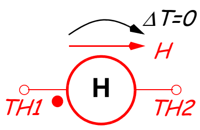
Name/Equation	Symbol and Nodes	Inputs and Outputs
Wattmeter (WM_TRB) POWER(t) = F(t) · v(t)		V [m/s]…Velocity F [N]…Force POWER [W]…Power
Displacement-Force-Representation—Rotational		
Angle Sensor (SM_ROTB) M(t) = 0		PHI [rad]…Angle
Torque Sensor (FM_ROTB) PHI(t) = 0 OMEGA(t) = 0		M [Nm]…Torque
Angular Velocity Sensor (VM_ROTB) M(t) = 0		OMEGA [rad/s]…Angular Velocity
Wattmeter (WM_ROTB) POWER(t) = M(t) · OMEGA(t)		OMEGA [rad/s]…Angular Velocity M [Nm]…Torque POWER [W]…Power
Velocity-Force-Representation—Translational_V		
Velocity Sensor (VM_TR) F(t) = 0		V [m/s]…Velocity
Force Sensor (FM_TR) Δs(t) = 0 Δv(t) = 0		F [N]…Force
Position Sensor (SM_TR) F(t) = 0 s(t=0) = S0		S0 [m]…Initial Position S [m]…Position
Wattmeter (WM_TR) POWER(t) = F(t) · v(t)		V [m/s]…Velocity F [N]…Force POWER [W]…Power

Name/Equation	Symbol and Nodes	Inputs and Outputs
Velocity-Force-Representation—Rotational_V		
Angular Velocity Sensor (FM_ROT) $M(t) = 0$		OMEGA [rad/s]…Angular Velocity
Torque Sensor (VM_ROT) $PHI(t) = 0 \quad OMEGA(t) = 0$		M [Nm]…Torque
Angle Sensor (SM_ROT) $M(t) = 0 \quad PHI(t=0) = PHI0$		PHI0 [rad]…Initial Angle PHI [rad]…Angle
Wattmeter (WM_ROT) $POWER(t) = M(t) \cdot OMEGA(t)$		OMEGA [rad/s]…Angular Velocity M [Nm]…Torque POWER [W]…Power

Thermal Meters

>>Basics>Measurement>Thermal

Thermal meters provide temperature (across quantity) and heat flow (through quantity) for thermal domain components.

Name/Equation	Symbol and Nodes	Outputs
Thermometer (THM) $H(t) = 0$		T [K]…Temperature
Heat Flow Sensor (HFM) $\Delta T(t) = 0$		H [J/s]…Heat Flow

8 Using Signal Characteristics

Evaluating the behavior of a dynamic system, there are several options to do it by special characteristic values. So also the influence of system parameters on the dynamic system behavior becomes more and more transparent.

For an automized system design it is necessary to have the interesting characteristic parameters already available during the simulation process. So an immediate adaption can be done. The State Graph Machines provide an excellent basis to observe or to respond on certain events within the system modelled.

Here are presented a couple of useful macros, once added to the users simulation model, determining various characteristic values. The macros are clearly arranged for different technical purposes: Common characteristic values, characteristic values for dynamic system behavior, characteristic values of dynamic performance of a controlled system or special wave form parameters.

Signal Characteristic components appear in the *Signal Characteristics* folder of the «Basics» tab. Select the folder *Signal Characteristics*, open *General Parameters*, *Dynamic Behavior Parameters*, *Dynamic Performance Parameters*, or *Special Waveform Parameters*, click a name, drag the component onto the sheet and release the mouse button.

You cannot use components in the *Signal Characteristic* folder in AC and DC simulations. To leave the components on the sheet, select the components and choose ELEMENT>DON'T ADD TO MODEL DESCRIPTION. When you want to process a transient simulation, you can change this option.

8.1 General Parameters

- Maximum/Maximum within Interval (MAX_PERIO/MIN_PERIO)
- Period Determination (TPERIO)
- Event Triggered Value (PROBE1)
- Event Triggered Calculation (PROBE2)

Maximum/Minimum within Interval

>>Basics>Signal Characteristics>General Parameters

The macro provides the maximum/minimum value of a input signal for a given interval. The beginning and end time of maximum/minimum determination TSTART and TSTOP have to be within'0' and the simulation end time TEND.

$$y(t) = \mathrm{MAX/MIN}(x(t)) \qquad \mathrm{TSTART} < t < \mathrm{TSTOP}$$

Dialog Settings

«Input Signal»
Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).

☑ ☐ **«Use Pin»**
If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

«Start Time [s]»«Stop Time [s]»
Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT	real
Measurement Start Time [s]	TSTART	real
Measurement Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Maximum/Minimum in Time Interval	VAL	real

Period Determination

>>Basics>Signal Characteristics>General Parameters

The macro provides the time interval between two successive, but otherwise independent upcoming events. Please note: The first event (condition) has to become true before the second. The first event (condition), must be true for beginning of timing; the second for end.

$$T = t(\mathrm{STOPEVENT}) - t(\mathrm{STARTEVENT})$$

Dialog Settings

«Start Event»«Stop Event»
Logical parameter type. Enter a numerical value, a variable, or an expression in the text box. The expression is interpreted as TRUE if the provided value is '1'. Otherwise the value is FALSE. The expression must be compliant to the SML syntax rules. See also “Equations, Expressions, and Variables” on page 32.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Start Event	STARTEVENT	real
Stop Event	STOPEVENT	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Period [s]	VAL	real

Event Triggered Value

>>Basics>Signal Characteristics>General Parameters

The macro provides the value of a variable, calculated when a specified event (condition) comes true. The logical expression for the event must conform to the syntax of the SIMPLORER expression interpreter.

$$VAL = x(EVENT) \quad \text{if} \quad EVENT{=}TRUE$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Event» Logical parameter type. Enter a numerical value, a variable, an expression in the text box. The expression is interpreted as TRUE if the provided value is '1'. Otherwise the value is FALSE. The expression must be compliant to the SML syntax rules. See also “Equations, Expressions, and Variables” on page 32.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT	real
Event for Measurement	EVENT	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value at Event	VAL	real

Event Triggered Calculation

>>Basics>Signal Characteristics>General Parameters

The macro block provides the values of two variables, calculated when two independent events (conditions) come true, and connecting them by an mathematical operation. Please note: The first event (condition) has to become true before the second. The first event (condition), must be true for beginning of timing; the second for end.

```
VAL_IN1 = x(EVENT1)        if  EVENT1=TRUE
VAL_IN2 = x(EVENT2)        if  EVENT2=TRUE
VAL = f(x(EVENT1),x(EVENT2),VARIABLES)
```


Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Event1»«Event2» Logical parameter type. Enter a numerical value, a variable, an expression in the text box. The expression is interpreted as TRUE if the provided value is '1'. Otherwise the value is FALSE. The expression must be compliant to the SML syntax rules. See also “Equations, Expressions, and Variables” on page 32.
	«Formula» Enter a mathematical expression in the text box. The expression must be compliant to the SML syntax rules. If you want to use the values at the event 1 and 2, you have to enter their qualifiers: `Name.VAL_IN1` and `Name.VAL_IN2`. The value of the operation is provided as output when EVENT2 becomes true. See also “Qualifier of Parameters” on page 29.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal 1 for Measurement 1	INPUT1	real
Input Signal 2 for Measurement 2	INPUT2	real
Event 1 for Measurement 1	EVENT1	real
Event 2 for Measurement 2	EVENT2	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Input Signal 1 at Event 1	VAL_IN1	real
Input Signal 2 at Event 2	VAL_IN2	real
Calculation Result	VAL	real

8.2 Dynamic Behavior Parameters

- Rise Time (TRISE)
- Fall Time (TFALL)
- Pulse Duration (TPULSE)
- Sensitivity

Rise Time

>>Basics>Signal Characteristics>Dynamic Behavior Parameters

The macro provides the rise time of a variable, in the case the final value of the variable in steady state has to be known. The rise time calculation starts at 10% of final value and ends at 90%. The rise time calculation is a non recurring operation within a specified interval.

Selecting a start time TSTART greater than zero, possibly occurring transient phenomena at the beginning can be suppressed. TSTART and TSTOP have to be within'0' and the simulation end time TEND.

$$
\begin{array}{ll}
t_{RISE} = t_2 - t_1 & TSTART < t < TSTOP \\
INPUT = x & DERIV = x' \\
t = t_1 & \text{if } \; x \geq 0 \wedge x' > 0 \wedge x = 0.1 \cdot FINALVAL \\
t = t_2 & \text{if } \; x \geq 0 \wedge x' > 0 \wedge x = 0.9 \cdot FINALVAL
\end{array}
$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Derivation of Input Signal» Enter name and qualifier of the first derivation of the input signal in the text box. Many SIMPLORER components provides this value, e.g. R1.dI, TRISE1.dVAL.
	«Maximum Value» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal (the rise time is to be obtained from)	INPUT	real
Derivative of Input Signal (for determination of rise or fall)	DERIV	real
Final Value of Input Signal in Steady State (signal in steady state)	FINALVAL	real

Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Rise Time (signal occurring in the given time interval) [s]	VAL	real

Fall Time

>>Basics>Signal Characteristics>Dynamic Behavior Parameters

The macro provides the fall time of a variable, in the case the final value of the variable in steady state has to be known. The fall time calculation starts at 90% of final value and ends at 10%. The rise time calculation is a non recurring operation within a specified interval

Selecting a start time TSTART greater than zero, possibly occurring transient phenomena at the beginning can be suppressed. TSTART and TSTOP have to be within'0' and the simulation end time TEND.

8

$$\begin{array}{ll} t_{FALL} = t_2 - t_1 & TSTART < t < TSTOP \\ INPUT = x & DERIV = x' \\ t = t_1 & \text{if} \quad x \geq 0 \wedge x' < 0 \wedge x = 0.1 \cdot FINALVAL \\ t = t_2 & \text{if} \quad x \geq 0 \wedge x' < 0 \wedge x = 0.9 \cdot FINALVAL \end{array}$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	**«Use Pin»** If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Derivation of Input Signal» Enter name and qualifier of the first derivation of the input signal in the text box. Many SIMPLORER components provides this value, e.g. R1.dI, TRISE1.dVAL.
	«Maximum Value» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal (the fall time is to be obtained from)	INPUT	real
Derivative of Input Signal (for determination of rise or fall)	DERIV	real
Final Value of Input Signal in Steady State (signal in steady state)	FINALVAL	real

Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Fall Time (signal occurring in the given time interval)	VAL	real

Pulse Duration

>>Basics>Signal Characteristics>Dynamic Behavior Parameters

The macro provides the pulse duration of a variable, in the case the final value of the variable has to be known. The calculation of pulse duration start at the rising edge, if the actual value of signal is >= 50% of final value. It ends at falling edge, if the actual value of signal decreases to <=50% of final value. The pulse duration calculation is a non-recurring operation within a specified interval.

Selecting a start time TSTART greater than zero, possibly occurring transient phenomena at the beginning can be suppressed. TSTART and TSTOP have to be within'0' and the simulation end time TEND.

$$t_{FALL} = t_2 - t_1 \qquad TSTART < t < TSTOP$$
$$INPUT = x \qquad DERIV = x'$$
$$t = t_1 \qquad \text{if} \quad x \geq 0 \wedge x' < 0 \wedge x = 0.5 \cdot FINALVAL$$
$$t = t_2 \qquad \text{if} \quad x \geq 0 \wedge x' > 0 \wedge x = 0.5 \cdot FINALVAL$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Derivation of Input Signal» Enter name and qualifier of the first derivation of the input signal in the text box. Many SIMPLORER components provides this value, e.g. R1.dI, TRISE1.dVAL.
	«Maximum Value» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal (the pulse duration is to be obtained from)	INPUT	real
Derivative of Input Signal (for determination of rise or fall)	DERIV	real
Final Value of Input Signal in Steady State (signal in steady state)	FINALVAL	real
Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Pulse Duration (signal occurring in the given time interval)	VAL	real

Sensitivity

>>Basics>Signal Characteristics>Dynamic Behavior Parameters

The C model provides the partial derivation of two input signals DERIV=δY/δX with the active values of the input signals at each simulation step. Within the dialog you can define the X range of values and a initial delay.

$$\mathrm{DERIV} = \frac{\partial y}{\partial x} = \frac{y(k) - y(k-1)}{x(k) - x(k-1)}$$

$$\mathrm{AVG} = \frac{1}{N} \cdot \sum_{k=1}^{N} \frac{\partial y}{\partial x} \qquad N = \text{Number of time step}$$

$$h = t(k) - t(k-1) \qquad h = \text{Current time step}$$

$$\mathrm{INPUTX} = x \qquad \mathrm{INPUTY} = y$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Delay [s]» Defines the initial delay of the sensitivity function. Enter a numerical value, a variable, or an expression in the text box to define the initial delay at simulation start.
	«X Interval Lower Limit»«X Interval Upper Limit» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

▼	**«Sample Time»** Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	**«Use Pin»** If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal Y for Derivation dy/dx	INPUTY	real
Input Signal X for Derivation dy/dx	INPUTX	real
Delay [s]	TDELAY	real
X-Interval Lower Limit	XMIN	real
X-Interval Upper Limit	XMAX	real
Sample Time [s]	TS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Derivative of Input Signal	DERIV	real
Mean Value of Derivative	AVG	real

8.3 Dynamic Performance Parameters

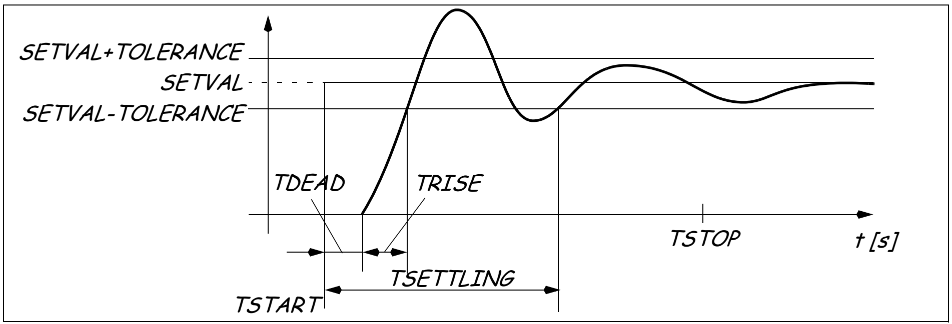
- Time Characteristics (TIME)
- Overshoot Characteristics (OVERSHOOT)
- Integral Characteristics (INTEGRAL)

Time Characteristics

>>Basics>Signal Characteristics>Dynamic Performance Parameters
The macro provides a possible dead time, rise time or settling time of 2nd order oscillating system. The macro is suitable to describe the dynamic behavior in case of setpoint changes or varying disturbances.

Selecting a start time `TSTART` greater than zero, the time calculation can be adapted to possibly occurring time shifts (Maybe a step change of setpoint occurs not before a certain time span has elapsed.). `TSTART` and `TSTOP` have to be within'0' and the simulation end time `TEND`.

For time definitions refer to the following figure:



Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Setpoint Value» Defines the setpoint of the input signal. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Tolerance» Defines the tolerance band around the setpoint (setpoint±tolerance). Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT	real
Setpoint	SETVAL	real
Tolerance	TOLERANCE	real
Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Dead Time [s]	TDEAD	real
Rise Time [s]	TRISE	real
Settling Time [s]	TSETTLING	real

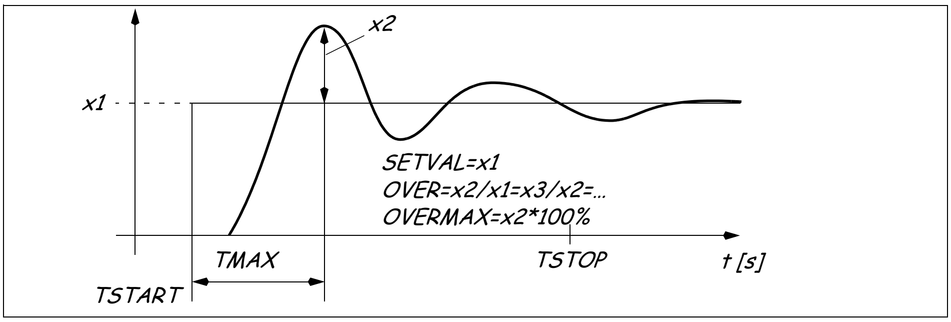
Overshoot Characteristics

>>Basics>Signal Characteristics>Dynamic Performance Parameters

The macro provides the overshoot factor, the maximum overshoot and the time span to reach the maximum overshoot. The maximum overshoot OVERMAX characterizes the system's ability to oscillate and is given in percent of final value.

Selecting a start time `TSTART` greater than zero, the time calculation can be adapted to possibly occurring time shifts (Maybe a step change of setpoint occurs not before a certain time span has elapsed.). Start time `TSTART` and end time `TSTOP` of rise time calculation have to be located within Zero and the whole simulation duration (`TEND`).

For time definitions refer to the following figure:



Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Setpoint Value» Defines the setpoint of the input signal. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Number of Overshoot Factors» Defines from which number of the overshoot ratio the calculation is started. Enter a positive integer in the text box.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal (Second reference signal, actual value)	INPUT	real
Setpoint (First reference signal, setpoint)	SETVAL	real

Number of Overshoot Factors for calculation	NFACT	real
Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Current Overshoot (Actual overshoot factor)	OVER	real
Maximal Overshoot (in percent of final value)	OVERMAX	real
Time of Maximal Overshoot (Time span to reach the maximum overshoot)	TMAX	real

Integral Characteristics

>>Basics>Signal Characteristics>Dynamic Performance Parameters

The macro provides integral criteria (also often called control Areas) of a variable. By means of such criteria the wave form of an error signal e(t) can be evaluated, caused by a setpoint change at $t = t_0$. Objective of the evaluation is an optimization of a controlled system by minimizing a criterion mentioned above.

Definition of Integral Criteria

For Control Areas without time weighting (IAE, ISE) the active values of an error signal at the beginning of the considered time interval have a big influence, otherwise later occurring deviations play only a minor role. For Control Areas with time weighting (ITAE, ITSE) the influence of the first part of signal is reduced significantly and later existing deviations are more taken into account. Using the time weighting criteria for controller optimization so produces better damped transient responses. The integral criteria are applicable for setpoint changes and also for changes in the disturbance signal.

$$IAE = \int |e(t)| dt \qquad ISE = \int e^2(t) dt$$

$$ITAE = \int t |e(t)| dt \qquad ITSE = \int t \cdot e^2(t) dt$$

$$INPUT = e$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Setpoint Value» Defines the setpoint of the input signal. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal (Second reference signal, actual value)	INPUT	real
Setpoint (First reference signal, setpoint)	SETVAL	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Deviation	DEV	real
Integral Absolute Error	IAE	real
Integral Squared Error	ISE	real
Integral Time Absolute Error	ITAE	real
Integral Time Squared Error	ITSE	real

8.4 Special Waveform Parameters

- Mean Value (MEAN)
- RMS Value (RMS)
- Sliding Mean Value (MEAN_SLD)
- Sliding RMS Value (RMS_SLD)
- Power and Energy (POWER)
- FFT (FFT)

Mean Value

>>Basics>Signal Characteristics>Special Waveform Parameters
The macro provides the mean value of a variable for a given interval. The `TSTART` and `TSTOP` parameters can be modified (mostly periodically) by variable assignments. So after each signal period an active mean value is available. The beginning and end time of maximum/minimum determination `TSTART` and `TSTOP` have to be within'0' and the simulation end time `TEND`.

$$\mathrm{MEAN} = \frac{1}{N} \cdot \sum_{k=1}^{N} x_k \qquad N = \text{Number of time step}$$

$$\mathrm{INPUT} = x \qquad \mathrm{VAL} = \mathrm{MEAN}$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal [/]	INPUT	real
Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Mean Value	VAL	real

RMS Value

>>Basics>Signal Characteristics>Special Waveform Parameters

The macro provides the RMS (root-mean-square) value of a variable for a given interval. The `TSTART` and `TSTOP` parameters can be modified (mostly periodically) by variable assignments. So after each signal period an actual mean value is available. The beginning and end time of maximum/minimum determination `TSTART` and `TSTOP` have to be within'0' and the simulation end time `TEND`.

$$\mathrm{RMS} = \sqrt{\frac{1}{N} \cdot \sum_{k=1}^{N} x_k^2} \qquad N = \text{Number of time step}$$

$$\mathrm{INPUT} = x \qquad \mathrm{VAL} = \mathrm{RMS}$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	**«Use Pin»** If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal [/]	INPUT	real

Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
RMS Value	VAL	real

Sliding Mean Value

>>Basics>Signal Characteristics>Special Waveform Parameters

The C Model provides the sliding mean value of a variable. The value is calculated within the user-defined time-window of the length TMEAS. The beginning and end time of value determination TSTART and TSTOP have to be within'0' and the simulation end time Tend. TMEAS should be the ten times the maximal simulation time step HMAX.

$$MEAN_{SLD} = \frac{1}{TMEAS} \cdot \int_{t}^{t+TMEAS} INPUT(t)dt \qquad TSTART \le t \le (TSTOP - TMEAS)$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (RLOAD.I, STEP.VAL, STATE.ST, VAR) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~N0002~~, ~~GND~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).
▼	«Sample Time» Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	«Use Pin» If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal [/]	INPUT	real
Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real
Integration Period [s]	TMEAS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Sliding Mean Value	VAL	real

Sliding RMS Value

>>Basics>Signal Characteristics>Special Waveform Parameters

The C Model provides the sliding mean value of a variable. The value is calculated within the user-defined time-window of the length `TMEAS`. The beginning and end time of value determination `TSTART` and `TSTOP` have to be within '0' and the simulation end time Tend. `TMEAS` should be ten times the maximal simulation time step `HMAX`.

$$RMS_{SLD} = \sqrt{\frac{1}{TMEAS} \cdot \int_{t}^{t+TMEAS} [INPUT(t)]^2 dt} \qquad TSTART \le t \le (TSTOP - TMEAS)$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	**«Use Pin»** If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).
▾	**«Sample Time»** Select **«System»** for simulator-defined variable sample time. Select **«Value»** to define a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity. Please note: The sample time must be a positive value.
☑ ☐	**«Use Pin»** If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal [/]	INPUT	real
Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real
Integration Period [s]	TMEAS	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Sliding RMS Value	VAL	real

Power and Energy

>>Basics>Signal Characteristics>Special Waveform Parameters
The macro provides the power and energy for a specified current and voltage within a given interval.

$$P = V \cdot I \qquad E = V \cdot I \cdot t$$

Dialog Settings	
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.
	«Start Time [s]»«Stop Time [s]» Common parameter type. Enter a numerical value, a variable, or an expression in the text box. The specified start time (for beginning of calculation) has to be less than the end time (for end of calculation).

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal Voltage	INPUT_V	real
Input Signal Current	INPUT_I	real
Start Time [s]	TSTART	real
Stop Time [s]	TSTOP	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Power	POWER	real
Energy	ENERGY	real

FFT

>>Basics>Signal Characteristics>Special Waveform Parameters

The model provides the harmonic analysis of a signal, in the case the fundamental frequency of the signal has to be known. The calculation is sine-based that means, a cosine signal would have a phase shift of $\pi/2$.

The signal, which is to investigate, is created from the input signal by so-called windows. Because of the multiplication with a window function all values outside of the window are '0'. Only the active window is used for the calculation. The different window functions can be considered as weighting factors for the signal within the window area.

Window function as Triangular, Von-Hann or Hamming should lower the leakage effect. In this case the sub-harmonics are split to frequencies next to them. The effect occurs if the chosen window range is not a integer of the multiples of the signal period. Not rectangular windows affect adversely the amplitude response.

Dialog Settings	
▼	«FFT Type» In the list, select **«Fixed»** or **«Flexible»**. Fixed means, only the first period (1/fundamental frequency is considered. Flexible means, each period is calculated again.
	«Delay [s]» Defines the initial delay before the first period begins. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Cut-off Frequency [Hz]» Defines the Cut-off frequency of the FFT analysis. Only harmonic waves with the frequency lower than the cut-off frequency are calculated. Common parameter type. Enter a numerical value, a variable, or an expression in the text box. Please note: The specification of '0' means there is no limitation.
	«Amplitude Filter[%]» Amplitudes smaller than the defined proportional value of the maximum are not displayed. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Window Type» In the list, select the window **«Rectangle»**, **«Triangular»**, **«Von-Hann»**, **«Hamming»**, **«Blackmann»**, **«Lanzcos»**, or **«Weber»**. Select the rectangular window if you want to investigate a periodical signal.
	«Fundamental Frequency [Hz]» Defines the fundamental frequency of the FFT analysis. The frequency corresponds to 1/TPERIO of the FFT model parameter. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Input Signal» Enter a quantity name used in your model sheet in the text box to define the input signal (`RLOAD.I`, `STEP.VAL`, `STATE.ST`, `VAR`) or use the pin to connect a quantity. Please note: You cannot use conservative nodes from circuits as input signal (~~`N0002`~~, ~~`GND`~~).
☑ ☐	«Use Pin» If the box is checked, the input signal is defined by the quantity connected to the pin, otherwise the value in the text box is used.

«Scanning Frequency [Hz]»
The scanning frequency corresponds to 1/TS of the FFT model parameter. The value should be at least twice greater than the highest analyze frequency. Enter a numerical value in the text box or use the pin to connect a quantity. Please note: The scanning frequency must be a positive value.

☑ ☐ «Use Pin»
If the box is checked, the sample time is defined by the quantity connected to the pin, otherwise the value in the text box is used.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	INPUT	real
Cut-off Frequency [Hz]	TS	real
Delay [s]	TDELAY	real
Period	TPERIO	real
FFT-type	TYPE	real
Window Type	WINDOW	real
Cut-off Frequency [Hz]	FCUT	real
Amplitude filter	AMPLFLT	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
0. … 23. Harmonic wave	AMPL[n]	real
Direct Component	AMPL[0]	real
0. … 23. Phase Angle in Radians [rad]	PHI[n]	real
0. … 23. Phase Angle in Degrees [deg]	PHIDEG[n]	real
0. … 23. Frequency in Hz	FREQ[n]	real
Fundamental Frequency [Hz]	FREQ[1]	real

9 Modeling Tools

Complex simulation models usually need simple auxiliary functions to model typical curves, characteristics, and equations simply and quickly. So the real subject of simulation can be investigated faster and more precisely.

Nonlinear components need a characteristic to represent a behavior, sources a time function to generate typical wave forms, and signals can be controlled dependent on their properties.

Modeling tools appear in the *Tools* folder of the «Basics» tab. The folder is subdivided in time functions, characteristics, and equations. Select the folder *Tools*, open *Time Functions*, *Characteristics*, or *Equations*, click a name, drag the component onto the sheet and release the mouse button.

All components can be used to provide input signals, characteristics, and mathematical functions for the simulation model.

9.1 Time Functions

- Sine Wave (SINE)
- Pulse Wave (PULSE)
- Triangular Wave (TRIANG)
- Trapezoidal Wave (TRAPEZ)
- Saw-tooth (ST_RISE/ST_FALL)
- Needle Pulses (NEEDLE)
- Arc Tangent (ARCTAN)
- Controlled Oscillator (VCO)
- 2D Lookup Table (LT_LIP/LT_NIP)
- PWM (/)

Using Time Functions

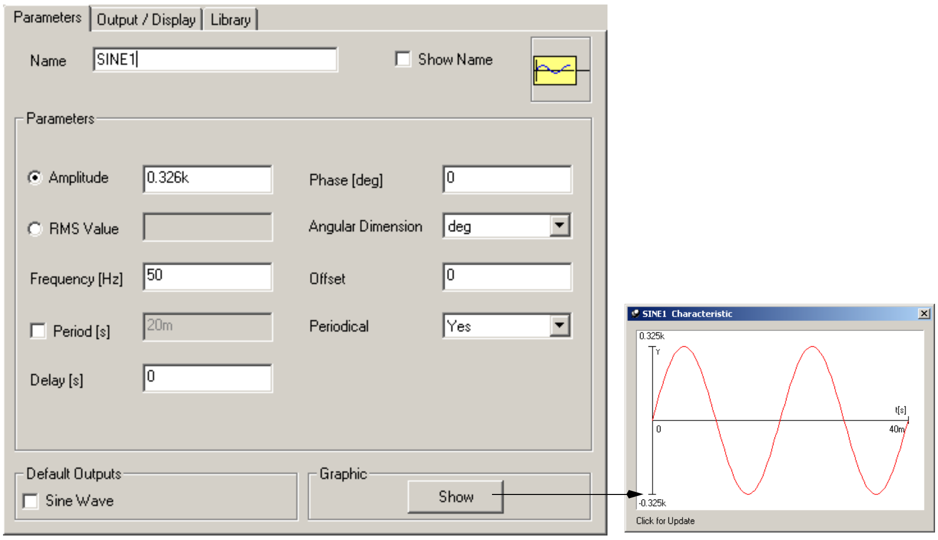
Time functions can be used for modeling in several ways.

- In electrical circuits, time functions can alter transient input signals of externally controlled sources.
- In block diagrams, time functions can define nominal values, disturbances and reference values.
- In the state graphs time functions define comparison functions and synchronization signals.

Predefined Time Functions

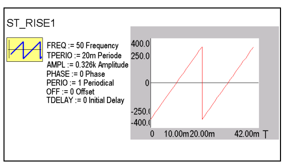
Predefined time functions have a quality determined behavior for which certain parameters can be defined. Predefined time functions are: Sine, Pulse, Triangular and Trapezoidal wave as well Needle pulses, Arc tangent and the Controlled oscillator.

In the property dialog, the frequency, period, initial delay, amplitude, phase, periodicity and offset can be modified. In addition, for all predefined time functions there is a preview mode, which displays the time functions for the defined points in a graphic window.

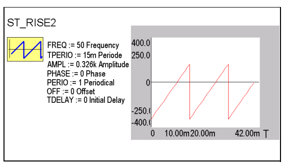


Changing Period

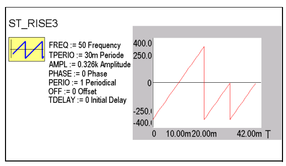
You can change the period independent of frequency. The period reduces or lengthens the signal depending on the period of the corresponding period (1/frequency).



Period = 1/f
the signal is depicted exactly



Period < 1/f
the signal is shortened



Period > 1/f
the signal is lengthened

User-Defined Time Functions

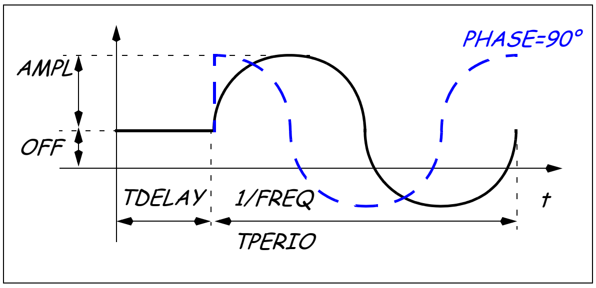
In addition to the predefined time functions, you can define your own time functions. The 2D Lookup Table create time function data pairs which are saved in an external file or in the simulation sheet.

The parameters period, initial delay, phase and periodicity can also be changed in this dialog. For all user-defined time functions there is a preview mode, which graphically displays the created time function.

Sine Wave

>>Basics>Tools>Time Functions

$$y(t) = \mathrm{AMPL} \cdot \sin(360° \cdot \mathrm{FREQ} \cdot t + \mathrm{PHASE}) + \mathrm{OFF}$$
$$\mathrm{RMS} = \mathrm{AMPL} \cdot \sqrt{2}$$



Dialog Settings	
⊙	«Amplitude» The absolute amplitude value is selected. Common parameter type. Enter a numerical value, a variable, or logical an in the text box.
⊙	«RMS» The RMS amplitude (amplitude•√2) value is selected. Common parameter type. Enter a numerical value, a variable, or logical an in the text box.
	«Frequency [Hz]» Defines the frequency of the sine function. Common parameter type. Enter a numerical value, a variable, or logical an in the text box.
☑ ☐	«Period [s]» If the box is checked, the period is defined by the value in the text box, otherwise the period is 1/frequency.
	«Delay [s]» Defines the initial delay of the sine function. Common parameter type. Enter a numerical value, a variable, or logical an in the text box.
	«Phase [deg/rad]» Defines the phase shift of the sine function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Angular Dimension» In the list, select radians or degrees. The dimension is applied to the phase.
	«Offset» Defines the phase offset of the sine function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Periodical» In the list, select **«Yes»** for a periodical sine function; select **«No»** for carrying out only one period corresponding to the defined frequency. After them the sine function is '0'.
	<Graphic> Click the button to display the sine function with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Frequency [Hz]	FREQ	real
Period [s]	TPERIO	real
Amplitude	AMPL	real
Phase [deg]	PHASE	real
Periodical	PERIO	real
Offset	OFF	real
Initial Delay [s]	TDELAY	real

Component Outputs

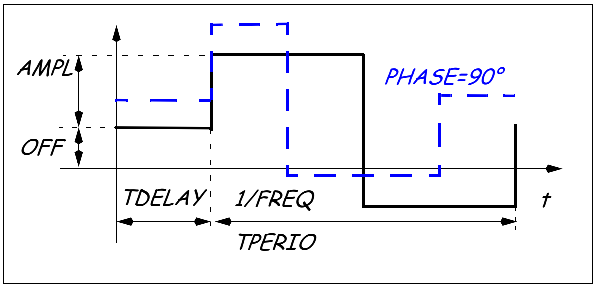
Description [Unit]	Parameter Name	Data Type
Sine Wave	VAL	real
Derivative of Value	dVAL	real

9

Pulse

>>Basics>Tools>Time Functions

$$y(t) = AMPL \cdot pulse(360° \cdot FREQ \cdot t + PHASE) + OFF$$



Dialog Settings	
	«Amplitude» Defines the amplitude value. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Frequency [Hz]» Defines the frequency of the pulse function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
☑ ☐	«Period [s]» If the box is checked, the period is defined by the value entered in the input filed, otherwise the period is 1/frequency.
	«Delay [s]» Defines the initial delay of the pulse function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

	«Phase [deg/rad]» Defines the phase shift of the pulse function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box. «Angular Dimension» In the list, select radians or degrees. The dimension is applied to the phase.
	«Offset» Defines the phase offset of the pulse function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Periodical» In the list, select **«Yes»** for a periodical pulse function; select **«No»** for carrying out only one period corresponding to the defined frequency. After them the pulse function is '0'.
	<Graphic> Click the button to display the pulse function with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Frequency [Hz]	FREQ	real
Period [s]	TPERIO	real
Amplitude	AMPL	real
Phase [deg]	PHASE	real
Periodical	PERIO	real
Offset	OFF	real
Initial Delay [s]	TDELAY	real

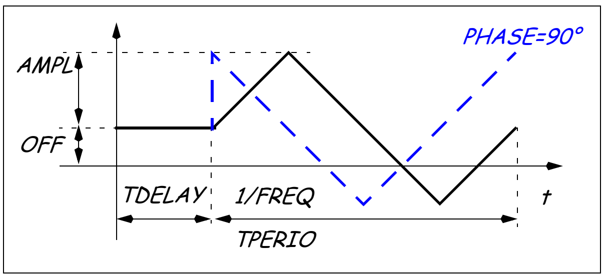
Component Outputs

Description [Unit]	Parameter Name	Data Type
Pulse	VAL	real
Derivative of Value	dVAL	real

Triangular Wave

>>Basics>Tools>Time Functions

$$y(t) = AMPL \cdot triang(360° \cdot FREQ \cdot t + PHASE) + OFF$$



Dialog Settings	
	«Amplitude» Defines the amplitude value. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Frequency [Hz]» Defines the frequency of the triangular function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
☑ ☐	«Period [s]» If the box is checked, the period is defined by the value entered in the input filed, otherwise the period is 1/frequency.
	«Delay [s]» Defines the initial delay of the triangular function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Phase [deg/rad]» Defines the phase shift of the triangular function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Angular Dimension» In the list, select radians or degrees. The dimension is applied to the phase.
	«Offset» Defines the phase offset of the triangular function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Periodical» In the list, select **«Yes»** for a periodical triangular function; select **«No»** for carrying out only one period corresponding to the defined frequency. After them the triangular function is '0'.
	<Graphic> Click the button to display the triangular function with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Frequency [Hz]	FREQ	real
Period [s]	TPERIO	real
Amplitude	AMPL	real
Phase [deg]	PHASE	real
Periodical	PERIO	real
Offset	OFF	real
Initial Delay [s]	TDELAY	real

Component Outputs

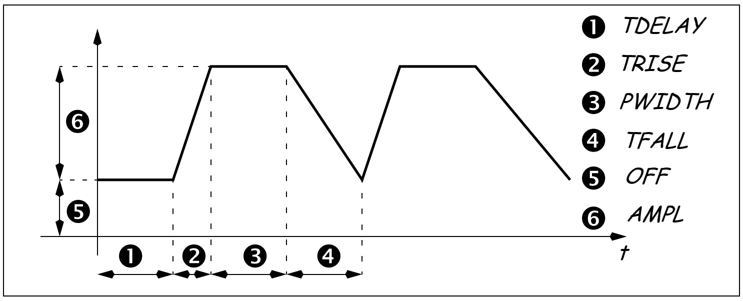
Description [Unit]	Parameter Name	Data Type
Triangular Wave	VAL	real
Derivative of Value	dVAL	real

Trapezoidal Wave

>>Basics>Tools>Time Functions

$$y(t) = AMPL \cdot trapez\left(360° \cdot \frac{1}{TPERIO} \cdot t + PHASE\right) + OFF$$

$$TPERIO = TRISE + TFALL + PWIDTH$$



Dialog Settings	
☑ ☐	«Period [s]» If the box is checked, the period is defined by the value entered in the input filed, otherwise the period is 1/frequency.
	«Delay [s]» Defines the initial delay of the trapezoidal function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
   ▼	«Phase [deg/rad]» Defines the phase shift of the trapezoidal function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box. «Angular Dimension» In the list, select radians or degrees. The dimension is applied to the phase.
	«Offset» Defines the phase offset of the trapezoidal function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Periodical» In the list, select «Yes» for a periodical trapezoidal function; select «No» for carrying out only one period corresponding to the defined frequency. After them the trapezoidal function is '0'.
	«Rise Time»«Fall Time»«Pulse Width» Defines the rise and fall time and the pulse width. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	<Graphic> Click the button to display the trapezoidal function with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Rise Time [s]	TRISE	real
Fall Time [s]	TFALL	real
Pulse Width [s]	PWIDTH	real
Period [s]	TPERIO	real
Amplitude	AMPL	real
Phase [deg]	PHASE	real
Periodical	PERIO	real
Offset	OFF	real
Initial Delay [s]	TDELAY	real

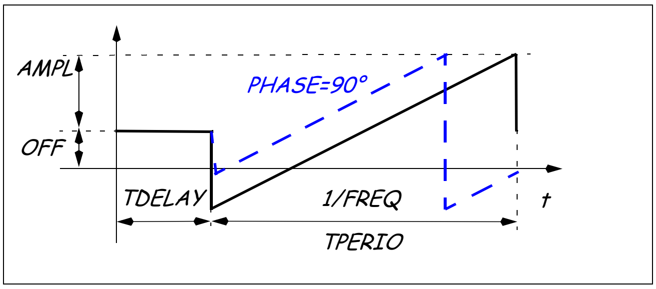
Component Outputs

Description [Unit]	Parameter Name	Data Type
Trapezoidal Wave	VAL	real
Derivative of Value	dVAL	real

Saw-tooth Function

>>Basics>Tools>Time Functions

$$y(t) = AMPL \cdot sawtooth(360° \cdot FREQ \cdot t + PHASE) + OFF$$



Dialog Settings	
▼	«Ramp Function» Select **«Rising»** for the saw-tooth rising or **«Falling»** for the saw-tooth falling function.
	«Amplitude» Defines the amplitude value. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Frequency [Hz]» Defines the frequency of the saw-tooth function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
☑ ☐	«Period [s]» If the box is checked, the period is defined by the value entered in the input filed, otherwise the period is 1/frequency.

	«Delay [s]» Defines the initial delay of the saw-tooth function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Phase [deg]» Defines the phase shift of the saw-tooth function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Angular Dimension» In the list, select radians or degrees. The dimension is applied to the phase.
	«Offset» Defines the Y shift of the saw-tooth function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	«Periodical» In the list, select **«Yes»** for a periodical saw-tooth function; select **«No»** for carrying out only one period corresponding to the defined frequency. After them the saw-tooth function is '0'.
	<Show> Click the button to display the saw-tooth function with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Frequency [Hz]	FREQ	real
Period [s]	TPERIO	real
Amplitude	AMPL	real
Phase [deg]	PHASE	real
Periodical	PERIO	real
Offset	OFF	real
Initial Delay [s]	TDELAY	real

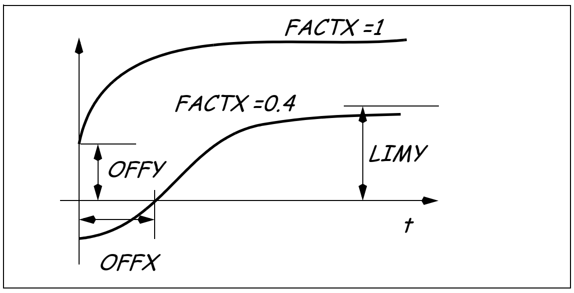
Component Outputs

Description [Unit]	Parameter Name	Data Type
Saw-tooth Function	VAL	real
Derivative of Saw-tooth Function	dVAL	real

Arc Tangent Function

>>Basics>Tools>Time Functions

$$y(t) = OFFY + \frac{LIMY}{\pi/2} \cdot \arctan(FACTX \cdot (t - OFFX))$$



Dialog Settings	
▼	«Angular Dimension» In the list, select radians or degrees. The dimension is applied to the Limit Y.
	«Limit Y [deg]» Defines the Y limit of the arc tangent function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Factor X» Defines the value of the factor X. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Offset X [s]»«Offset Y» Defines the X/Y shift of the arc tangent function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Graphic Time» Defines the value of the time (X axis) to display the arc tangent function in the preview.
	<Show> Click the button to display the arc tangent function with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Limit Y [deg]	LIMY	real
Factor X	FACTX	real
Offset X [s]	OFFX	real
Offset Y	OFFY	real

Component Outputs

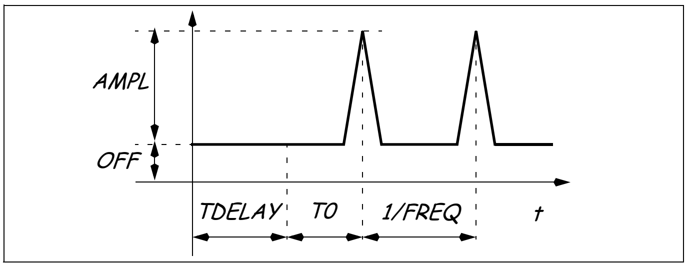
Description [Unit]	Parameter Name	Data Type
Arc Tangent Function	VAL	real

Needle Pulses

>>Basics>Tools>Time Functions

The function provides needle pulses with a defined amplitude and frequency after a delay to the simulation start and a delay within the period at the output.

$$y(t) = AMPL \quad \text{if} \quad t = TDELAY + T0 + \frac{K}{FREQ} \qquad K \in N$$



Dialog Settings	
	«Amplitude» Defines the amplitude value. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Frequency [Hz]» Defines the frequency of the needle pulses. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Delay [s]» Defines the initial delay of the needle pulses. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Delay within Period» Defines the delay of the needle pulses within the period. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Offset» Defines the phase offset of the needle pulses. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
▼	**«Periodical»** In the list, select **«Yes»** for a periodical needle pulses; select **«No»** for carrying out only one period corresponding to the defined frequency. After them the trapezoidal function is '0'.
	<Graphic> Click the button to display the trapezoidal function with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Frequency [Hz]	FREQ	real
Amplitude of Pulses	AMPL	real

Delay within the Period [s]	T0	real
Periodical	PERIO	real
Offset	OFF	real
Initial Delay [s]	TDELAY	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Needle Pulse Function	VAL	real

Controlled Oscillator

>>Basics>Tools>Time Functions

$$
\begin{array}{lll}
y(t) = \mathrm{AMPL} \cdot \sin(360° \cdot f + \mathrm{PHASE}) & & \\
f = f(\mathrm{CTRL}) & & \\
f = \mathrm{UL} & \text{when} & \mathrm{UL} < \mathrm{CTRL} \\
f = \mathrm{LL} & \text{when} & \mathrm{LL} > \mathrm{CTRL}
\end{array}
$$

Dialog Settings	
☐	«Limit Input Signal» Each value of the input signal, which controls the frequency of the output signal, is accepted.
☑	«Limit Input Signal» If the box is checked, you can enter an upper and lower limit in the corresponding text box. Values outside of the range are ignored. «Upper Limit»«Lower Limit» Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Output Amplitude» Defines the amplitude of the sine wave. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Phase» Defines the phase shift of the sf the created sine wave. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
	«Angular Dimension» In the list, select radians or degrees. The dimension is applied to the phase.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Input Signal	CTRL	real
Input Signal Upper Limit	UL	real
Input Signal Lower Limit	LL	real
Amplitude	AMPL	real
Phase [deg]	PHASE	real

Component Outputs

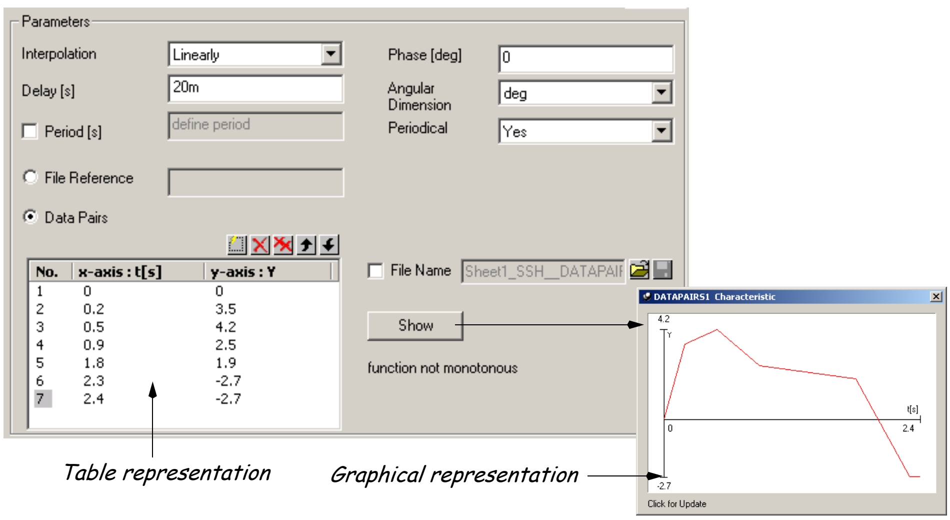
Description [Unit]	Parameter Name	Data Type
Sine Wave	VAL	real
Current Frequency [Hz]	FREQ	real

2D Lookup Table

>>Basics>Tools>Time Functions

The function allows the definition of wave forms from a set of fixed data points with linear interpolation between them (straight lines from point to point) or rectangular lines between them (two orthogonal lines from point to point, which are parallel to the coordinate axes).

The X values of the data-pairs must be monotonous rising. The last slope is effective for all values outside the X range. If you want to have a constant value outside the X range, you have to define two data-pairs with the same Y value at the end.



The time T values in the data file are be matched exactly, because of the simulator's internal step size algorithms. The next value of T=T+dt are used for the computation.

The simulator tries to match all values of a data set exactly. Very large data sets (e.g., from an measuring instrument) cause a reduction in the simulator speed. If possible, reduce the data sets to a required minimum to decrease simulation time.

The function LOOKUP(X,Y) provides the Y value of a given X value. LOOKUP(XY1.VAL,5) -> Y value of the characteristic XY1 for the X value 5.

2D Lookup Table with Interpolation

$$y(t) = y(i) + \frac{y(i+1) - y(i)}{t(i+1) - t(i)} \quad \text{t lies in the i-th interval}$$

2D Lookup Table without Interpolation

$$y(t) = y(i) \quad \text{for} \quad t(i) \le t \le (i+1) \quad \text{t lies in the i-th interval}$$

Dialog Settings	
▼	«Interpolation» In the list, select **«Linearly»** for straight lines from point to point of the fixed data set points; select **«Without»** for two orthogonal lines from point to point, which are parallel to the coordinate axes.
	«Delay [s]» Defines the initial delay of the 2D lookup table. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.
☑ ☐	«Period [s]» If the box is checked, the period is defined by the value entered in the text box, otherwise the period is defined from the last X value in the lookup table.
   ▼	«Phase [deg]» Defines the phase shift of the 2D lookup table function. Common parameter type. Enter a numerical value, a variable, or an expression in the text box. «Angular Dimension» In the list, select radians or degrees. The dimension is applied to the phase.
▼	«Periodical» In the list, select **«Yes»** for a periodical 2D table function; select **«No»** for carrying out only one period corresponding to the defined data pairs and period. After them the function is continued with the last slope.
☐	«File Name» If no file name is used, you have to define data-pairs in the table on the right side. Create, delete, or move entries with the symbols placed on the upper right side. The time (X value) must be monotonous rising. See also “Defining Parameters” on page 49.
☑	«File Name» To define the data points, a two-dimensional data file is used which can be created by the DAY Post Processor or a text editor. The XY-data file always has two channels (columns) where time (X value) and magnitude (Y value) are stored. The length (number of lines) of the data files is arbitrary. To open or save files, click the symbols next to the text box. Please note: Select the correct file filter (.mdx) before browsing. See also “The .mdx File Format” on page 467.
	<Show> Click the button to display the 2D lookup table with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Period	TPERIO	real
Phase [deg]	PHASE	real
Periodical	PERIO	real
Initial Delay [s]	TDELAY	real
File Name	FILE	file

Component Outputs

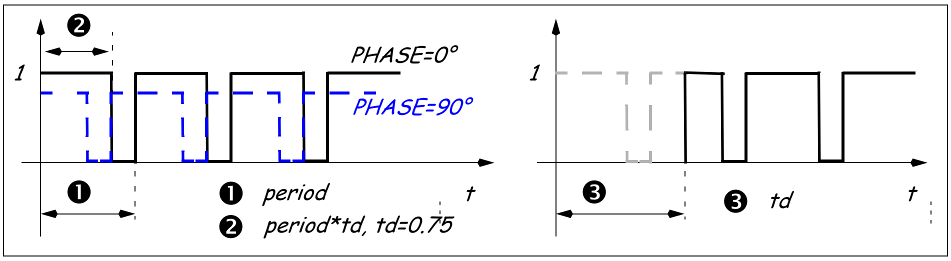
Description [Unit]	Parameter Name	Data Type
Value	VAL	real

PWM

>>Basics>Tools>Time Functions

The PWM model provides a pulse-width modulated signal with amplitude '1'. The value remains at '1' for a fraction duty time of the period time. Each duty time ≥1 results to an amplitude '1'. The phase parameter shifts the positive edge of the PWM. If the phase shift is set to '0', the PWM value goes up to '1' with the start of every new period. The initial delay value delays the PWM at simulation start. As long as the simulation time is smaller than the initial delay, the output of the PWM remains '0'. Please note: The delay behavior of the PWM time functions differs from the other time functions.

The minimum time step should consider the PWM times. Especially the synchronization on the edges of the PWM depends on the selected `HMIN`.



Dialog Settings	
	«Value» Click the «Value» field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below.
	«Name»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Period Time [s]	period	real
Duty Cycle [/]	dc	real
Phase Shift [deg]	phase	real
Initial Delay [s]	td	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
PWM Output [/]	val	real

9.2 Characteristics

- Equivalent Line (EQUL)
- Exponential Function (EXP)
- Hyperbolic Function (HYP)
- Diffusion Capacitance (CDI)
- Junction Capacitance (CSP)
- Polynom of 2nd Order (PO2)
- 2D Lookup Table (XY)
- 3D Lookup Table (/)

Using Characteristics

Similar to time functions, characteristics can also be used to model nonlinear characteristics. In the electric circuit module, nonlinear characteristics are preferable for the behavioral description of passive components, static or dynamic behavior of semiconductor devices and controlled sources. In the block diagram module, the nonlinear transfer behavior of blocks can be described.

The characteristics are placed in the *Tools* folder in the «Basics» tab. Select the folder, click *Characteristics*, select a component, drag the element onto the sheet and release the mouse button.

Predefined Characteristics

Predefined characteristics have a quality determined behavior for which certain parameters can be defined. Predefined characteristics are: Equivalent Line, Exponential Function, Hyperbolic Function and Polynom of 2nd Order as well Diffusion Capacitance and Junction Capacitance.

User-Defined Characteristics

In addition to the predefined characteristics, SIMPLORER users can define their own characteristic. The 2D Lookup Table XY create characteristic data pairs which are saved in an external file or in the simulation sheet.

Equivalent Line

>>Basics>Tools>Characteristics

V_{limit} stands for the intersection of the forward and reverse characteristic and is calculated by the program itself. With the equivalent line the nonlinear behavior of switching components is represented most simply. However, in some rare circumstances numeric instabilities can occur with the equivalent line when switching from one branch to the other.

$$I = \frac{v(t) - VF}{RB} \quad \text{for} \quad v(t) \geq V_{limit}$$

$$I = \frac{v(t)}{RR} \quad \text{for} \quad v(t) < V_{limit}$$

Dialog Settings
«Forward Voltage [V]»«Bulk Resistance [Ohm]»«Reverse Resistance [Ohm]» Defines the coefficients forward voltage, bulk resistance, and reverse resistance. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Forward Voltage [V]	VF	real
Bulk Resistance [Ohm]	RB	real
Reverse Resistance [Ohm]	RR	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value of the Equivalent Line	VAL	real

Exponential Function

>>Basics>Tools>Characteristics

With the exponential function, the static behavior of switched components is represented accurately over a wide range. However, the use of exponential functions without specifying a bulk resistance RB is not very practical for semiconductor components in power electronics. For these cases, the bulk resistance must be inserted by the user as a separate component in the circuit.

$$i(t) = ISAT \cdot \left(e^{\frac{v(t)}{VT}} - 1\right) \quad \text{for} \quad v(t) \geq 0$$

$$i(t) = \frac{v(t)}{RR} \quad \text{for} \quad v(t) < 0$$

Dialog Settings
«Thermal Voltage [V]»«Saturation Current [A]»«Reverse Resistance [Ohm]» Defines the coefficients thermal voltage, saturation current, and reverse resistance. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Thermal Voltage [V]	VT	real
Saturation Current [A]	ISAT	real
Reverse Resistance [Ohm]	RR	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value of the Exponential Function	VAL	real

Hyperbolic Function

>>Basics>Tools>Characteristics

Because of the great sensitivity of the hyperbolic parameters caused by the large difference in bulk and reverse resistance, SIMPLORER calculates these parameters itself. In contrast to the equivalent line function, the hyperbolic function has a steady first derivation in the entire range of input values. Thus, convergence problems possible with the equivalent line are avoided.

$$i(t) = V_1 \cdot v(t) + V_2 + \sqrt{V_3 \cdot (v(t))^2 + V_4 \cdot v(t) + V_5}$$

Dialog Settings
«Forward Voltage [V]»«Bulk Resistance [Ohm]»«Reverse Resistance [Ohm]» Defines the coefficients forward voltage, bulk resistance, and reverse resistance. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Forward Voltage [V]	VT	real
Bulk Resistance [Ohm]	RB	real
Reverse Resistance [Ohm]	RR	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value	VAL	real

Diffusion Capacitance

>>Basics>Tools>Characteristics

The diffusion capacitance represents the storage of the minority carrier within the junction after forward current flow. The diffusion and the junction capacitance were developed for modeling the dynamic switching behavior of the diode or another semiconductor component.

$$q(t) = (TAU \cdot ISAT) \cdot \left(e^{\frac{v(t)}{VT}} - 1\right)$$

$$C_{Diff} = \frac{dq(t)}{dv(t)} = TAU \cdot \frac{ISAT}{VT} \cdot e^{\frac{v(t)}{VT}}$$

Dialog Settings
«Thermal Voltage [V]»«Saturation Current [A]»«Minority Carrier Lifetime [s]» Defines the coefficients thermal voltage, saturation current, and minority carrier lifetime. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Thermal Voltage [V]	VT	real
Saturation Current [A]	ISAT	real
Minority Carrier Lifetime [s]	TAU	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value of the Diffusion Capacitance	VAL	real

Junction Capacitance

>>Basics>Tools>Characteristics

The junction capacitance represents the dependence of the resulting junction capacitance from the applied voltage due to the majority carrier distribution. The junction and the diffusion capacitance were developed for modeling the dynamic switching behavior of a diode or another semiconductor component.

$$C = \frac{C_0}{\sqrt{1 - \frac{V}{V_d}}}$$

Dialog Settings
«Zero-bias Junction Capacitance [F]»«Diffusion Voltage [V]» Defines the coefficients zero-bias junction capacitance and diffusion voltage. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Zero-bias Junction Capacitance [F]	C0	real
Diffusion Voltage [V]	VD	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value of the Junction Capacitance	VAL	real

Polynom of 2nd Order (parabolic)

>>Basics>Tools>Characteristics

V_{limit} stands for the intersection of forward and reverse characteristics and is calculated by the program itself. The parabolic function is characterized by a very favorable convergence behavior and thus is well suited for modeling frequently switched systems. There are approximations which show a good coincidence between model and real components using all three parameters, a, b and c. In obtaining these parameters, however, special care must be taken so that only sensible combinations are specified. Parameters a, b and c are freely definable coefficients. Experience has proven that one should follow these guidelines in determining the parabolic parameters:

$$i(t) = A + B \cdot v(t) + C(v(t))^2 \quad \text{for} \quad v(t) \geq V_{limit}$$

$$i(t) = \frac{v(t)}{RR} \quad \text{for} \quad v(t) < V_{limit}$$

$$A = 0 \qquad B = \frac{1}{RR} \qquad C = \frac{I_{fav}}{V(I_{fav})^2} \qquad R = 100k\Omega$$

Dialog Settings

«Reverse Resistance [Ohm]»
Defines the coefficient reverse resistance. Common parameter type. Enter a numerical value, a variable, or an expression in the text box.

«Coefficient A/B/C»
Defines the coefficient A, B and C. Common parameter types. Enter a numerical value, a variable, or an expression in the text box.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Reverse Resistance [Ohm]	RR	real
Coefficient A/B/C	A/B/C	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value of the Polynom	VAL	real

2D Lookup Table XY

>>Basics>Tools>Characteristics

The 2D Lookup Table XY is used to describe nonlinear behavior different from the predefined characteristics. To achieve a simple and universal embedding of characteristics for describing nonlinear behavior of components, characteristics can also be given through XY data pairs. It is possible to create data files with:

- the component dialog
- the DAY Post Processor, or
- a text editor.

The function LOOKUP(X,Y) provides the Y value of a given X value. LOOKUP(XY1.VAL,5) -> Y value of the characteristic XY1 for the X value 5.

The X-Y assignment of current and voltage (as physical quantities) depends on the application of characteristics in the simulation model. See also 3.4 “Characteristics in Simulation Models” on page 55.

Dialog Settings	
	«Unit X-Axis»«Y-Axis» Defines the units for X and Y Axis. The units are displayed in the column headings in the look-up table and saved in the created .mdx file.
☐	«File Name» If no file is used, enter the data-pairs in the table on the right side. Create, delete, or move entries with the symbols placed on the upper right side. See also “Defining Parameters” on page 49.
☑	«File Name» To define the data points, a two-dimensional data file is used which can be created by the DAY Post Processor or a text editor. The XY data file always has two channels (columns) where X value and Y value are stored. The length (number of lines) of the data files is arbitrary. To open or save files, click the symbols next to the text box. Please note: Select the correct file filter (.mdx) before browsing. See also “The .mdx File Format” on page 467.
	<Show> Click the button to display the 2D lookup table with the defined values. Please note: If you use variables for the definition, the preview is empty.

Component Parameters

Description [Unit]	Parameter Name	Data Type
File Name	FILE	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value	VAL	real

Components Using Characteristics

The mapping of the X and Y values to physical quantities (voltage, current, and so on) depends on the used characteristics within the simulation model. The table below shows how SIMPLORER interprets the data to calculate the present value of a component.

Component	Name	x	y
RNL	Nonlinear resistor	v	i
GVNL	Nonlinear voltage-controlled conductor	v	G
CNL	Nonlinear capacitor	q	v
CVNL	Nonlinear dual capacitor	v	C
LNL	Nonlinear inductor	Ψ	i
LINL	Nonlinear dual inductor	i	L
D	Diode	v	i
BJT	Transistor	v	i
TH	Thyristor	v	i
GTO	GTO-Thyristor	v	i
TRIAC	TRIAC	v	i
IGBT	IGBT	v	i
MOS	MOS	v	i
EVNL, IVNL	Nonlinear voltage-controlled sources	vc	E/I
EINL, IINL	Nonlinear current-controlled sources	ic	E/I
NL	Nonlinear transfer block	input	val
DCMENL	DC Machine Nonlinear electrical excitation	i	Ψ/L

Symbol	Description [Unit]
v	Component voltage [V]
i	Component current [A]
Ψ	Flux linkages of the inductor [Vs]
q	Charge of the capacitor [As]
L	Inductance [H]
C	Capacitance [F]
vc	Control voltage [V]
ic	Control current [A]
input	Block input (without dimension)
val	Block output (without dimension)
E/I	Source voltage/Source current

Separate Component Characteristic

The separate component characteristic dialog is used if you define a characteristic directly in the dialog of components using nonlinear characteristics.

Characteristic Dialog Settings	
⊙	«Reference» The characteristic of the component is defined by an *2D Lookup Table XY* component. Enter the name in the text box. See also “2D Lookup Table XY” on page 295.
⊙	«Lookup Table» The lookup table option is selected. Select **«File Reference»** or **«Data Pairs»**.
⊙	«File Reference» The file reference option is selected. Enter a variable name, referring to a file name, in the text box.
⊙	«Data Pairs» If no file is used, enter the data-pairs in the table. Create, delete, or move entries with the symbols placed on the upper right side. See also “Defining Parameters” on page 49.
	<Show> Click the button to display the 2D lookup table with the defined values. Please note: If you use variables for the definition, the preview is empty. Click in the dialog window to refresh the representation when values were changed in the table view.
	«Monotony» If X values are not monotonous rising, the text **«Function not monotonous»** is displayed.
☑	«File Name» To define the data points, a two-dimensional data file is used which can be created by the DAY Post Processor or a text editor. The XY data file always has two channels (columns) where X value and Y value are stored. The length (number of lines) of the data files is arbitrary. To open or save files, click the symbols next to the text box. Please note: Select the correct file filter (.mdx) before browsing. See also “The .mdx File Format” on page 467.

3D Lookup Table

>>Basics>Tools>Characteristics

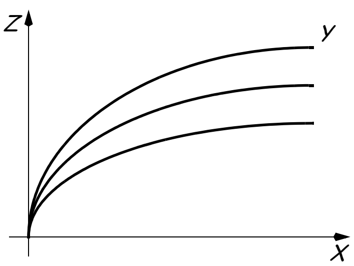
The 3D Lookup Table is used from components to describe nonlinear behavior different from the predefined characteristics, for example a temperature-dependent resistor. The 3D Lookup Table uses a special .mdx file which describes a parameter-dependent family of characteristics; during the simulation the component determines the required values by interpolation or extrapolation from the values in the .mdx file. The characteristics can are given through XYZ Data sets. It is possible to create data files with:

- the component dialog
- the DAY Post Processor, or
- a text editor.

The .mdx file is composed of characteristic blocks with 3 channels. For each parameter value (which represents, for example, a temperature) the values of the corresponding characteristic are defined. See also “The .mdx File Format” on page 467.

Requirements for Characteristic Blocks

- a physically useful characteristic for each parameter value (for voltage '0' is also current '0', otherwise errors occur during simulation)
- the same sequence of monotonously increased or decreased values for the X axis
- a constant select parameter value (Y axis) for each block
- across all blocks the select parameter must be monotonously increased or decreased.



X	Z	Y	
0	0	0	*1. Block*
1	1	0	*Characteristic for value 0 (Y)*
2	2	0	
0	0	1	*2. Block*
1	2	1	*Characteristic for value 1 (Y)*
2	4	1	
0	0	2	*3. Block*
1	3	2	*Characteristic for value 2 (Y)*
2	6	2	

Using 3D Lookup Tables with Components

Open the property dialog of a nonlinear component, for example of a resistor. Select **«Nonlinear»** in the dialog and click <Characteristic>. Enter the name of the 3D Lookup Table component in the text box (Name.VAL, for example *Werte3D501.VAL)*. You cannot use the pin function with the 3D Lookup Table.

During simulation, the current values of the nonlinear component are available for X,Y, and Z value in the 3D Lookup Table component. If you define outputs, you can display these quantities.

Components Using Characteristics

The table shows SIMPLORER's interpretation of the corresponding values of nonlinear components. The X, Y, and Z axis are interpreted as follows:

Component	Description	x-Axis 1. Channel	z-Axis 2. Channel	y-Axis 3. Channel
RNL	nonlinear resistance	voltage v [V]	current i [A]	Parameter
GVNL	nonlinear conductance	current i [A]	voltage v [V]	Parameter
LNL	nonlinear inductance	flux Ψ [Vs]	current i [A]	Parameter
LINL	nonlinear dual inductance	current i [A]	inductance L [H]	Parameter
CNL	nonlinear capacitance	charge Q [As]	voltage v [V]	Parameter
CVNL	nonlinear dual capacitance	voltage v [V]	capacitance C [F]	Parameter

See also 3.4 "Characteristics in Simulation Models" on page 55.

Dialog Settings

«Parameter Variable»
The variable value selects one of the defined characteristic blocks. Enter a variable name in the text box. The range of the values must be available in the characteristic file; otherwise, incorrect values are used.

▼	«Interpolation» In the list, select **«Linear»** or **«Spline»**. Within the characteristic the values are interpolated; outside the values are extrapolated. The spline interpolation is only available if the X and Y value is at least 4 points away from the limit. Therefore, the spline interpolation is useful only if the characteristic has more than 8x8 data points. If the characteristic is smaller than 8x8, linear interpolation is used, even if spline interpolation is defined.
☐	«File Name» If no file is used, enter the datasets in the tables. Create, delete, or move entries with the symbols placed on the upper right side. See also “Defining Parameters” on page 49.
☑	«File Name» To define the data points, a three-dimensional data file is used which can be created by the DAY Post Processor or a text editor. The XZY data file always has three channels (columns) where X, Y, and Z value are stored. The length (number of lines) of the data files is arbitrary. To open or save files, click the symbols next to the text box. Please note: Select the correct file filter (.mdx) before browsing. See also “The .mdx File Format” on page 467.
☑ ☐	«Unit X Axis»«Unit Y Axis»«Unit Z Axis» Defines the units for X, Y, and Z Axis. Check the options and enter the unit in the text box. The units are saved in the created .mdx file.
	«X Axis» Defines the X values of the each characteristic block. Create, delete, or move entries with the symbols placed on the upper right side. If you click <Sweep Data>, you can sweep data for a given range of values. <Set> replaces the values in the lookup table, <Append> inserts the new data at the table end.
	«Y Axis» Defines the select value of the characteristic block. The value is taken from the value of the parameter variable. Create, delete, or move entries with the symbols placed on the upper right side and enter a numerical values in the table. If you click <Sweep Data>, you can sweep data for a given range of values. <Set> replaces the values in the lookup table, <Append> inserts the new data at the table end.
	«Z Axis» Defines the Z values for each X value. Click a Y value in the mid-table. All available X values are displayed in the right table. Click in the **«Z axis»** field and enter a numerical value. See also requirements for characteristic blocks.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Interpolation Method, 1-linear, 2-spline	Interpolation	real
Variable Name of Select Parameter	FamilyParameter	real
File Name, 3-dimensional .mdx File	FamilyOfCurves	file

Component Outputs

Description [Unit]	Parameter Name	Data Type
Value	VAL	real
Current X, Y, Z Values of the Nonlinear Component	X, Y, Z	real

9.3 Equations

- Initial Values (IVA)
- Equations (EQU)
- DES (DES)

Initial Values

>>Basics>Tools>Equations

Variables defined in an Initial Value Assignment is computed **once** at the simulation start. They can be used in each component on the sheet.

Dialog Settings	
	«Expression» Create, delete, or move entries with the symbols placed on the upper right side. Choose **«Equation»** to enter the expression in the single line mode. Choose **«If…Else…»** to open the equation editor window. In the single line mode, click in the **«Expression»** field and define the expression, for example `C:=A+B`, `var1:=sin(2*PI*50*t)`. See also “Defining Parameters” on page 49. and “Equations, Expressions, and Variables” on page 32.
…	«Expression» To open the equation editor window, double-click the symbol in the text box or choose **«If… else…»** on the drop-down menu.
	The editor window can display the complete text of an equation. In addition, you can insert standard mathematical functions and If-Else statements from a panel. To insert functions, click the f(x) symbol and insert a function by clicking one of the symbols on the panel. To define an If-Else statement click the new entry symbol and select one of the statements. Define conditions and variable assignments in the text to complete the inserted template. To check the syntax of an expression, click the ✓ symbol. Errors and warnings are displayed below the edit box. Click <OK> to close the equation editor window and return to the single line mode. If errors in the definition occur, the ⊗ symbol is displayed at the beginning of an entry in the list. See also “Equations, Expressions, and Variables” on page 32.
	«Info» Field for user specific information about an expression. Click in the field of the corresponding entry and enter the text. You can display these information on the sheet by defining the display setting within the **«Output/Displays»** tab.
☑ ☐	«Output» If the box is checked, the variable is saved in the result file (.sdb) during the simulation. Please note: If you want to display a variable value, you have to select the box in the output dialog or in a Display Element. If the box is deactivated, you cannot display the variable value after the simulation. See also “Defining Displays and Outputs” on page 51.

Equations

>>Basics>Tools>Equations

The equation function provides mathematical expressions to all SIMPLORER modules. The formula interpreter manages general mathematical and logical expressions, which consist of a number of operands and operators. In addition, widely used mathematical standard functions are supported. Variables are allowed as operands (for example voltages or currents from the circuit components, quantities from the Block Diagram and variables and states from the State Graph). Of course, variables from other expressions can also be used in an expression. The name of a variable is defined by the user. It can be up to 50 characters long and must start with a letter.

The spelling of the variable names is case sensitive.

Dialog Settings	
	«Expression» Create, delete, or move entries with the symbols placed on the upper right side. Choose **«Equation»** to enter the expression in the single line mode. Choose **«If…Else…»** to open the equation editor window. In the single line mode, click in the **«Expression»** field and define the expression, for example `C:=A+B`, `var1:=sin(2*PI*50*t)`. See also “Defining Parameters” on page 49. and “Equations, Expressions, and Variables” on page 32.
	«Expression» To open the equation editor window, double-click the symbol in the text box or choose **«If…else…»** on the drop-down menu.
	The editor window can display the complete text of an equation. In addition, you can insert standard mathematical functions and If-Else statements from a panel. To insert functions, click the f(x) symbol and insert a function by clicking one of the symbols on the panel. To define an If-Else statement click the new entry symbol and select one of the statements. Define conditions and variable assignments in the text to complete the inserted template. To check the syntax of an expression, click the ✓ symbol. Errors and warnings are displayed below the edit box. Click <OK> to close the equation editor window and return to the single line mode. If errors in the definition occur, the ⊗ symbol is displayed at the beginning of an entry in the list. See also “Equations, Expressions, and Variables” on page 32.
	«Calculation Sequence» In the list, select **«Standard»**, **«Before State Graph»**, or **«After State Graph»**. Standard means the equation is calculated first of all simulation quantities (before circuit, block diagram, state graph, time functions, …). Before State Graph is computed **before** the state graph is processed; After State Graph, **after** the state graph is processed. See also 13.6 “Simulator Backplane” on page 418.
	«Info» Field for user specific information about an expression. Click in the field of the corresponding entry and enter the text. You can display these information on the sheet by defining the display setting within the **«Output/Displays»** tab.

☑ ☐	«Output» If the box is checked, the variable is saved in the result file (.sdb) during the simulation. Please note: If you want to display a variable value, you have to select the box in the output dialog or in a Display Element. If the box is deactivated, you cannot display the variable value after the simulation. See also "Defining Displays and Outputs" on page 51.

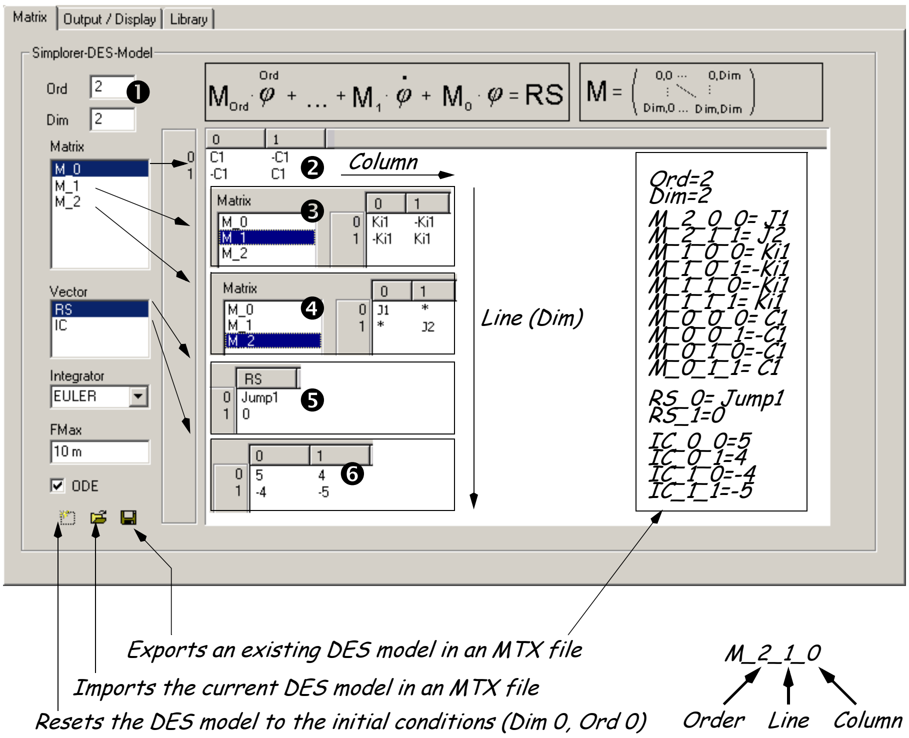
DES Solver

>>Basics>Tools>Equations

The DES model integrates differential equation systems independent from electrical nodes and without equivalent circuits for circuits and block diagrams. Order and dimension of the equation system are unlimited. The component cannot be used with AC and DC simulation.

DES models are solved in a separate differential equation system. The simulation parameters HMIN and HMAX from the TR simulator are valid for the DES model too. The DES model is calculated after the circuit module. See also 13.6 "Simulator Backplane" on page 418.

Please note: If you use the Runge-Kutta algorithm, the matrix M with the highest order must be non-singular.



1	Order and dimension of the equation system
2 3 4	Definition of the matrix coefficients
5	Definition of the vector on the right hand side
6	Definition of the initial values of the solution vector

In empty fields the * must be placed; 0 is interpreted as a numerical value.

Equations

For the common equation:

$$M\_2 \cdot \ddot{\varphi} + M\_1 \cdot \dot{\varphi} + M\_0 \cdot \varphi = RS$$

the results with the inserted coefficients of the example are:

$$\begin{bmatrix} J_1 & * \\ * & J_2 \end{bmatrix} \cdot \begin{bmatrix} \dfrac{d^2\varphi_1}{dt^2} \\ \dfrac{d^2\varphi_2}{dt} \end{bmatrix} + \begin{bmatrix} Ki_1 & -Ki_1 \\ -Ki_1 & Ki_1 \end{bmatrix} \cdot \begin{bmatrix} \dfrac{d\varphi_1}{dt} \\ \dfrac{d\varphi_2}{dt} \end{bmatrix} + \begin{bmatrix} C_1 & -C_1 \\ -C_1 & C_1 \end{bmatrix} \cdot \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} = RS$$

Dialog Settings	
	«Ord»«Dim» Defines order and dimension of the equation system. Enter a numerical value (non-negative integer with or without unit suffix) in the text box to define the order and positive integer to define the dimension. Please note: You cannot define a variable or expression in these cases. Only a number is accepted.
	«Matrix» Defines the vector coefficients. Select a matrix entry and click in the fields of the corresponding coefficient in the matrix dialog on the right hand side. Common parameter type. Enter a numerical value, a variable, or an expression.
RS	«Vector» Defines the coefficients of the vector on the right hand side. Select **«RS»** and click in the fields of the corresponding coefficient in the matrix dialog on the right hand side. Common parameter type. Enter a numerical value, a variable, or an expression.
IC	«Vector» Defines the coefficients of the initial values of the solution vector. Select **«IC»** and click in the fields of the corresponding coefficient in the matrix dialog on the right hand side. Common parameter type. Enter a numerical value, a variable, or an expression.
▼	«Integrator» Defines the integration algorithm of the DES model. The algorithm is independent of the integration algorithm of the main simulator. If **«ODE»** is checked, you can select also the Runge-Kutta algorithms in the list. These algorithms expect a non-singular matrix for the matrix *M* with the highest order. If the matrix does not meet the condition, the simulator uses the **«Euler»** integration method for calculating.
	«Fmax» Defines the accuracy of the right side computation of the differential equation system. Enter a numerical value (with or without unit suffix) in the text box. Please note: The simulator tries to meet the defined maximum error as long as the minimum time step of the higher simulator is not reached. If `HMIN` is reached and Fmax is larger than the defined value, the simu.lator goes to the next step despite the deviation.

☑ ☐	**«ODE»** ODE means Ordinary Differential Equation. If the box is checked, the Runge-Kutta Algorithm is available as integration method, otherwise only **«Euler»** and **«Trapez»** is available in the integrator list box.
	«New» Resets the DES model to the initial conditions (Dim 0, Ord 0).
	«Import/Export MDX File» To define the matrix coefficients, a data file can be used. The file have to correspond with form as shown in the example above. To open or save files, click one of the symbols. Please note: Select the correct file filter (.mtx) before browsing. See also "The .mdx File Format" on page 467.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Order of Equation System	ORD	real
Dimension of Equation System	DIM	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Coefficients of the Solution Vector	SV[dim,ord]	real

10 Physical Domains

Physical Domain components appear in the *Physical Domains* folder of the «Basics» tab. The folder is subdivided in fluidic, magnetic, mechanical, and thermal domains. Select the folder *Physical Domains*, open one of the folders, click a name, drag the component onto the sheet and release the mouse button.

Along with electrical domain components you can model and simulate systems consisting of different engineering applications. Each domain has its own relation for across and through quantities. In addition, components and wires of a domain are represented by an identical color in the graphical representation of the Schematic. To change the settings of nature colors, choose SHEET>PROPERTIES. Please note: You can only change the wiring color, not the color of component symbols.

All models are implemented as C models.

The *Physical Domains* folder provides components for the following domains:

- Fluidic Components
- Magnetic Components
- Mechanical Components
- Thermal Components

Mechanical components are subdivided into four physical domains: translational – rotational and translational_v – rotational_v

You cannot use components in the *Physical Domains* folder in AC and DC simulations. To leave components on the sheet, select the components and choose ELEMENT>DON'T ADD TO MODEL DESCRIPTION. When you want to process a transient simulation, you can change this option.

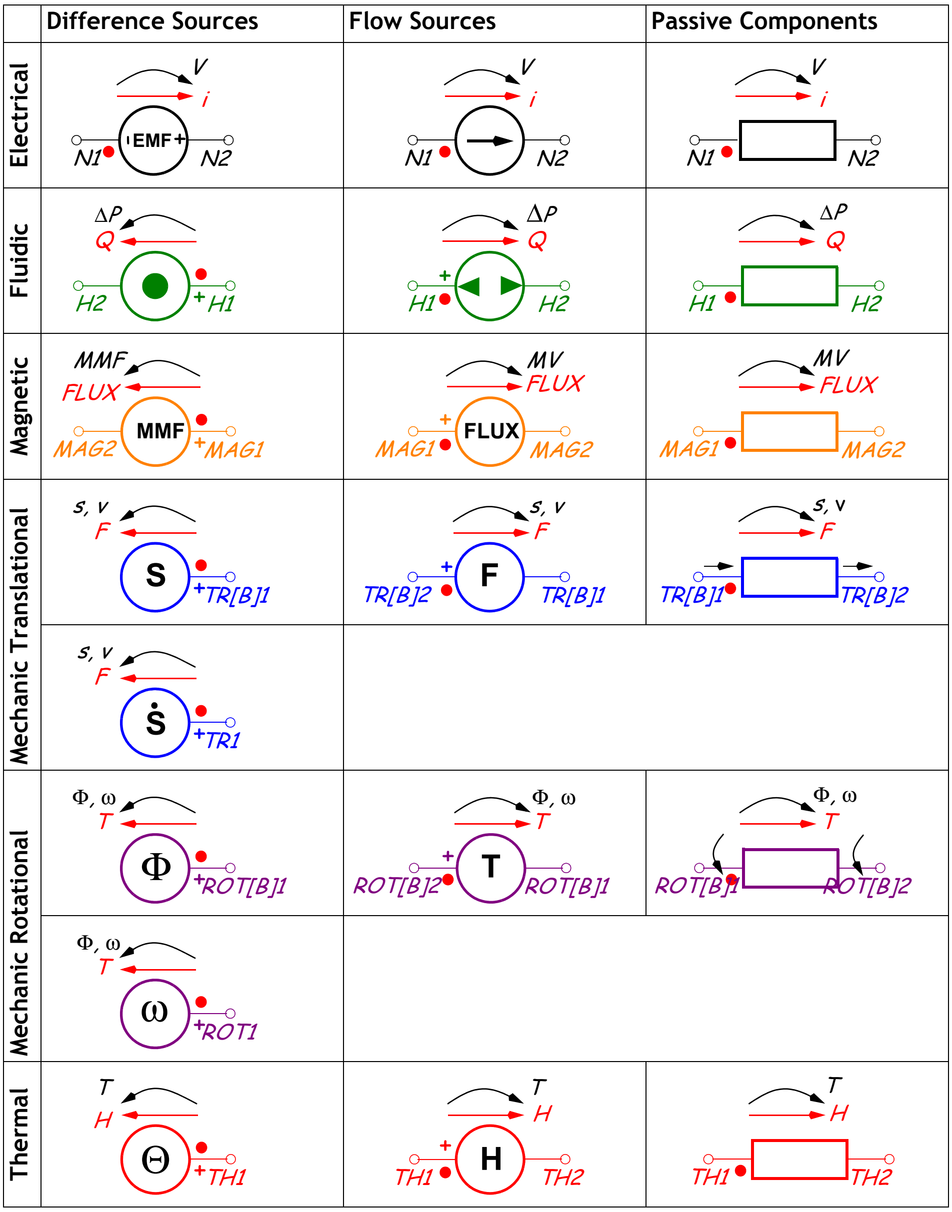
Load Reference Arrow System of Physical Domains

The table shows the reference arrow system for all in SIMPLORER available domains.

The counting direction is marked by the red point at the component symbol. The red point is always at pin 1 of a component.

Across quantities: Value=Value at Pin1 – Value at Pin 2.

Through quantities: Value is positive if the quantity flows at the red point into the component.



10

Nature Types of Components

Nature types are properties from conservative nodes (ports) of components. At least one specific nature exists for each domain. The mechanical domain has four nature types.

In the Schematic, you can connect conservative nodes only if they have the same nature type. By way of contrast, other quantities (non-conservative nodes) can be changed between all domains.

The library in the «Basic» tab provides components with the following nature types:

- electrical
- fluidic
- magnetic
- translational
- translational_v
- rotational
- rotational_v
- thermal

Connecting Components

Connections between components are created in the wire mode (CTRL+W or CONNECT>WIRE). Place the cross wire on a component pin on the sheet, and set the beginning, corners, and end of the wire with the mouse. Press ESC to finish the wire mode.

Conservative component nodes must always be connected with a node of the same nature type. To connect conservative nodes of different nature types you must use the D2D (domain to domain) component. See also Domain to Domain Transformation component in the «Tool» tab.

Schematic tests the pins during the routing and will not allow invalid connections (for example between an electrical and magnetic nature).

After the simulator starts, Schematic tests the complete simulation model. If connections are missing, an error message or warning is displayed in the «Build» area of the Information window. Double-click the warning or error text to move the component with the wrong connection into the visible range of the Schematic.

10.1 Fluidic Components

- Pressure Source (P)
- Flow Source (Q)
- Hydraulic Resistance (RHYD)
- Hydraulic Capacitance (CHYD)
- Hydraulic Inductance (LHYD)

Across and Through Quantity of the Fluidic Domain

Across	Through	Equation
Pressure [Pa]	Flow Rate [m^2/s^2]	$q(t) = K \cdot p(t)$

The fluidic domain components appear in the *Fluidic* folder of *Physical Domains* in the «Basics» tab. The folder provides fluidic basic components to model simple fluidic equivalent circuits. The fluidic domain uses the pressure as across and the flow rate as through quantitate.

By default, components and wires of the fluidic domain are represented in green in the graphical representation of the Schematic. Select SHEET>PROPERTIES to change the wire color of the domain. You can only change the wire color. The symbol color is not affected by the setting.

Pressure Source

>>Basics>Physical Domains>Fluidic

The component represents an ideal pressure source which supplies the defined pressure differential for an arbitrary flow rate. Within the dialog you can choose a constant or the time-controlled type.

Dialog Settings	
⊙	«Pressure Differential [Pa]» The linear pressure source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⊙	«Time Controlled» The time-controlled pressure source is selected. In the list, select a time function and define the corresponding parameters. See also “Using Time Functions” on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Pressure Source

$$\Delta p(t) = P = VALUE$$

The pressure is equivalent to the present value of the quantity specified in the text box. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The flow through an ideal pressure source can be arbitrary. Real pressure sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Pressure Differential [Pa]	VALUE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Pressure Differential [Pa]	P	real
Flow Rate [m^3/s^2]	Q	real

Component Nodes

Description	Node Name	Nature Type
Node with high potential (with red point)	H1	fluidic
Node with low potential	H2	fluidic

Time-Controlled Pressure Source

$\Delta p(t) = f[\text{time function}]$

The time controlled pressure sources have the same properties as the linear pressure source above; the flow rate can be arbitrary and the internal resistance is zero (ideal source). The value of the pressure differential instead is equivalent to the present value of the selected time function at each calculation step.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [Pa]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	real
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Pressure Differential [Pa]	P	real
Flow Rate [m^3/s^2]	Q	real

Component Nodes

Description	Node Name	Nature
Node with high potential (with red point)	H1	fluidic
Node with low potential	H2	fluidic

Flow Source

>>Basics>Physical Domains>Fluidic

The component represents an ideal flow source which supplies the defined flow rate for an arbitrary pressure differential. Within the dialog you can choose a constant or the time-controlled type.

Dialog Settings	
⊙	«Flow Rate [m^3/s^2]» The linear flow source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⊙	«Time Controlled» The time-controlled flow source is selected. In the list, select a time function and define the corresponding parameters. See also “Using Time Functions” on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Flow Source

```
q(t) = VALUE
```

The flow rate is equivalent to the present value of the quantity specified in the input field. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The pressure of an ideal flow source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Flow Rate [m^3/s^2]	VALUE	real

Simulator Outputs

Description [Unit]	Parameter Name	Data Type
Pressure Differential [Pa]	P	real
Flow Rate [m^3/s^2]	Q	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	H1	fluidic
Pin 2	H2	fluidic

Time-Controlled Flow Source

$$q(t) = f[\text{time function}]$$

The time controlled flow sources have the same properties as the linear flow source above; the pressure can be arbitrary and the internal resistance is zero (ideal source). The value of the flow rate instead is equivalent to the present value of the selected time function at each calculation step.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [m^3/s^2]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Pressure Differential [Pa]	P	real
Flow Rate [m^3/s^2]	Q	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	H1	fluidic
Pin 2	H2	fluidic

Hydraulic Resistance

>>Basics>Physical Domains>Fluidic

The component represents a simple fluid resistance specified by a linear coefficient.

$$q(t) = K \cdot p(t)$$

Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Linear Coefficient [$m^3/(s*Pa)$]	K	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Differential Pressure [Pa]	P	real
Flow Rate [m^3/s]	Q	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	H1	fluidic
Pin 2	H2	fluidic

Hydraulic Capacitance

>>Basics>Physical Domains>Fluidic

The component represents a hydraulic resistance that also includes the effects of fluid compressibility, or hydraulic capacitance. The characteristics are based on equations for laminar and turbulent flow through a smooth round pipe supplemented with the differential equation for effect of fluid compressibility.

$$q(t) = \frac{dp}{dt} \cdot \frac{VOL}{B} + P0$$

Dialog Settings	
↓↑	**«Value»** Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. **«Name»«Default»«Unit»«Description»** Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Fluid Volume [m^3]	VOL	real
Bulk Modulus [Pa]	B	real
Initial Pressure [Pa]	P0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Differential Pressure [Pa]	P	real
Flow Rate [m^3/s]	Q	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	H1	fluidic

Hydraulic Inductance

>>Basics>Physical Domains>Fluidic

The component represents a hydraulic resistance that also includes the effects of fluid inertia, or hydraulic inductance. The characteristics are based on equations for laminar and turbulent flow through a smooth round pipe supplemented with the differential equation for effect of fluid mass and acceleration.

$$A = \frac{\pi \cdot DIA^2}{4} \qquad L_{HYD} = \frac{RHO \cdot LEN}{A} + Q0 \qquad P(t) = L_{HYD} \cdot \frac{dq}{dt}$$

Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Fluid Density [kg/m³]	RHO	real
Diameter [m]	DIA	real
Length [m]	LEN	real
Initial Flow Rate [m³/s]	Q0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Differential Pressure [Pa]	P	real
Flow Rate [m³/s]	Q	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	H1	fluidic
Pin 2	H2	fluidic

10.2 Magnetic Components

- Magneto-Motive Force Source (MMF)
- Flux Source (FLUX)
- Magneto-Resistor (RMAG)
- Winding (ELTOMAG)

Across and Through Quantity of the Magnetic Domain

Across	Through	Equation
Magneto-Motive Force [A]	Magnetic Flux [Vs]	$MV(t) = K \cdot FLUX(t)$

The magnetic domain components appear in the *Magnetic* folder of *Physical Domains* in the «Basics» tab. The folder provides magnetic basic components to model simple magnetic equivalent circuits. The magnetic domain uses the magneto-motive force as across and the magnetic flux as through quantity.

By default, components and wires of the magnetic domain are represented in orange in the graphical representation of the Schematic. Select SHEET>PROPERTIES to change the wire color of the domain. You can only change the wire color. The symbol color is not affected by the setting.

Magneto-Motive Force Source

>>Basics>Physical Domains>Magnetic

The component represents an ideal magneto-motive force source which supplies the defined magnetic force for an arbitrary magnetic flux. Within the dialog you can choose a constant or time-controlled source.

Dialog Settings	
⊙	«Magneto-Motive Force [A]» The linear magneto-motive force source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⊙	«Time Controlled» The time-controlled magneto-motive force source is selected. In the list, select a time function and define the corresponding parameters. See also “Using Time Functions” on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Magneto-Motive Force Source

```
MV(t) = VALUE
```

The magneto-motive force is equivalent to the present value of the quantity specified in the text box. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The magnetic flux of an ideal force source can be arbitrary. Real force sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Magneto-Motive Force [A]	VALUE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Magneto-Motive Force [A]	MV	real
Magnetic Flux [Vs]	FLUX	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	MAG1	magnetic
Pin 2	MAG2	magnetic

Time-Controlled Magneto-Motive Force Source

$MV(t) = f[\text{time function}]$

The time controlled force sources have the same properties as the linear force source above; the flux can be arbitrary and the internal resistance is zero (ideal source). The value of the force instead is equivalent to the present value of the selected time function at each calculation step.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [A]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Magneto-Motive Force [A]	MV	real
Magnetic Flux [Vs]	FLUX	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	MAG1	magnetic
Pin 2	MAG2	magnetic

Magnetic Flux Source

>>Basics>Physical Domains>Magnetic

The component represents an ideal magnetic flux source which supplies the defined magnetic flux for an arbitrary magneto-motive force. Within the dialog you can choose a constant or time-controlled source.

Dialog Settings	
⊙	«Magnetic Flux [Vs]» The linear magnetic flux source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⊙	«Time Controlled» The time-controlled flux source is selected. In the list, select a time function and define the corresponding parameters. See also "Using Time Functions" on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Magnetic Flux Source

```
FLUX(t) = VALUE
```

The magnetic flux is equivalent to the present value of the quantity specified in the text box. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The magneto-motive force of an ideal flux source can be arbitrary. Real flux sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Magnetic Flux [Vs]	VALUE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Magnetic Voltage [A]	MV	real
Magnetic Flux [Vs]	FLUX	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	MAG1	magnetic
Pin 2	MAG2	magnetic

Time-Controlled Magnetic Flux Source

$$FLUX(t) = f[\text{time function}]$$

The time controlled flux sources have the same properties as the linear flux source above; the force can be arbitrary and the internal resistance is zero (ideal source). The value of the flux rate instead is equivalent to the present value of the selected time function at each calculation step.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [Vs]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Magnetic Voltage [A]	MV	real
Magnetic Flux [Vs]	FLUX	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	MAG1	magnetic
Pin 2	MAG2	magnetic

Magneto-Resistor

>>Basics>Physical Domains>Magnetic

The component represents a magneto-resistor that means, a homogeneous distributed magnetic field within a media of proportional B-H-relation to model air gaps and leakage paths.

$$MV(t) = K \cdot FLUX(t)$$

Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Linear Coefficient [A/(V*s)]	K	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Magnetic Voltage [A]	MV	real
Magnetic Flux [V*s]	FLUX	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	MAG1	magnetic
Pin 2	MAG2	magnetic

Winding

>>Basics>Physical Domains>Magnetic

The component represents a coupling element between electric and magnetic circuit. It converts electrical energy into magnetic energy and vice versa (bidirectional energy flow). The component has electrical pins, N1 and N2 (across: v, through: i) and magnetic pins, MAG1 and MAG2. You can specify the number of turns (coupling quantity between magnetic and electric domain) and the ohmic winding resistance.

$$v(t) = -W \cdot \frac{d}{dt}FLUX + R \cdot i(t) \qquad MV(t) = W \cdot i(t)$$

Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Resistance [Ω]	R	real
Number of Turns	W	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Voltage [V]	V	real
Current [A]	I	real
Magnetic Voltage [A]	MV	real
Magnetic Flux [V*s]	FLUX	real

Component Nodes

Description	Node Name	Nature
Positive node Pin 1 (with red point)	N1	electrical
Pin 2	N2	electrical
Positive node Pin 1 (with red point)	MAG1	magnetic
Pin 2	MAG2	magnetic

10.3 Mechanical Components

Displacement-Force-Representation

Translational	Rotational
• Position Source (S_TRB)	• Angle Source (S_ROTB)
• Velocity Source (V_TRB)	• Angular Velocity Source (V_ROTB)
• Force Source (F_TRB)	• Torque Source (F_ROTB)
• Spring (SPRING_TRB)	• Spring (SPRING_ROTB)
• Damper (DAMP_TRB)	• Damper (DAMP_ROTB)
• Mass (MASS_TRB)	• Mass (MASS_ROTB)
• Spacer (SPACER_TRB)	• Spacer (SPACER_ROTB)
• Limit Stop (LIMIT_TRB)	• Limit Stop (LIMIT_ROTB)

Velocity-Force-Representation

Translational	Rotational
• Velocity Source (V_TR)	• Angular Velocity Source (V_ROT)
• Force Source (F_TR)	• Torque Source (F_ROT)
• Spring (SPRING_TR)	• Spring (SPRING_ROT)
• Damper (DAMP_TR)	• Damper (DAMP_ROT)
• Mass (MASS_TR)	• Mass (MASS_ROT)
• Limit Stop (LIMIT_TR)	• Limit Stop (LIMIT_ROT)

Across and Through Quantities of Mechanical Domains

Representation	System	Across	Through	Equation
Displacement-Force	Translational	s [m]	F [N]	$v(t) = \frac{ds}{dt}$
	Rotational	φ [rad]	M [Nm]	$\omega(t) = \frac{d\varphi}{dt}$
Velocity-Force	Translational	v [m/s]	F [N]	$s(t) = \int v dt$
	Rotational	ω [rad/s]	M [Nm]	$\varphi(t) = \int \omega dt$

The mechanical domain components appear in the *Mechanical* folder of *Physical Domains* in the «Basics» tab. The folder provides mechanical basic components to model simple mechanical equivalent circuits. There are two representation forms: the displacement-force representation and the velocity-force representation. The table shows the across and through quantities used of the representation forms.

By default, components and wires of the mechanical domain are represented in blue for translational systems and in lilac for rotational systems. Select SHEET>PROPERTIES to change the wire color of the domain. You can only change the wire color. The symbol color is not affected by the setting.

Position/Angle Source

>>Basics>Physical Domains>Mechanical>Displacement-Force-Representation
The component represents an ideal mechanical position source which supplies the defined position for an arbitrary force/torque.

Dialog Settings	
⦿	«Displacement [m]»«Angle [rad]» The linear temperature source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⦿	«Time Controlled» The time-controlled temperature source is selected. In the list, select a time function and define the corresponding parameters. See also “Using Time Functions” on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Position/Angle Source

s(t) = VALUE   φ(t) = VALUE

The displacement/angle is equivalent to the present value of the quantity specified in the text box. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The force/torque through an ideal position/angle source can be arbitrary. Real force/torque sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	VALUE/VALUE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin Position Source	TRB1/TR1	translational/ translational_v
Pin Angle Source	ROTB1/ROT1	rotational/ rotational_v

Time-Controlled Position/Angle Source

$s(t) = f[\text{time function}]$ $\varphi(t) = f[\text{time function}]$

The time controlled position/angle sources have the same properties as the linear position/angle source above; the force/torque flow can be arbitrary and the internal resistance is zero (ideal source). The value of the position/angle instead is equivalent to the present value of the selected time function at each calculation step.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [m]/[rad]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin Position Source	TRB1/TR1	translational/ translational_v
Pin Angle Source	ROTB1/ROT1	rotational/ rotational_v

Velocity/Angular Velocity Source

>>Basics>Physical Domains>Mechanical>Displacement-Force-Representation

>>Basics>Physical Domains>Mechanical>Velocity-Force-Representation

The component represents an ideal mechanical velocity source which supplies the defined velocity for an arbitrary force/torque.

Dialog Settings	
⦿	«Velocity [m/s]»«Angular Velocity [rad/s]» The linear velocity source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⦿	«Time Controlled» The time-controlled velocity source is selected. In the list, select a time function and define the corresponding parameters. See also “Using Time Functions” on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Velocity Source

$$\frac{ds}{dt} = v(t) = VALUE \qquad \frac{d\varphi}{dt} = \omega(t) = VALUE$$

The velocity is equivalent to the present value of the quantity specified in the text box. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The force/torque through an ideal position/angle source can be arbitrary. Real force/torque sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Velocity [m/s]/Angular Velocity [rad/s]	VALUE/VALUE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin Velocity Source	TRB1/TR1	translational/ translational_v
Pin Angular Velocity Source	ROTB1/ROT1	rotational/ rotational_v

Time-Controlled Velocity Source

$$\frac{ds}{dt} = v(t) = f[\text{time function}]$$

$$\frac{d\varphi}{dt} = \omega(t) = f[\text{time function}]$$

The time controlled velocity sources have the same properties as the linear velocity source above; the force/torque flow can be arbitrary and the internal resistance is zero (ideal source). The value of the position/angle instead is equivalent to the present value of the selected time function at each calculation step.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [m/s]/[rad/s]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin Velocity Source	TRB1/TR1	translational/ translational_v
Pin Angular Velocity Source	ROTB1/ROT1	rotational/ rotational_v

Force/Torque Source

>>Basics>Physical Domains>Mechanical>Displacement-Force-Representation

>>Basics>Physical Domains>Mechanical>Velocity-Force-Representation

The component represents an ideal mechanical force/torque source which supplies the defined force/torque for an arbitrary position.

Dialog Settings	
⦿	«Force [N]»«Torque [Nm]» The linear force/torque source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⦿	«Time Controlled» The time-controlled force/torque source is selected. In the list, select a time function and define the corresponding parameters. See also “Using Time Functions” on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Force/Torque Source

$$F(t) = VALUE \qquad M(t) = VALUE$$

The force/torque is equivalent to the present value of the quantity specified in the input field. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The position of an ideal force/torque source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Force [N]/Torque [Nm]	VALUE/VALUE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin1-Pin2 Force Source	TRB1-TRB2/TR1-TR2	translational/ translational_v
Pin1-Pin2 Torque Source	ROTB1-ROTB2/ROT1-ROT2	rotational/ rotational_v

Time-Controlled Force/Torque Source

$$F(t) = f[\text{time function}] \qquad M(t) = f[\text{time function}]$$

The time controlled force/torque sources have the same properties as the linear force/torque source above; the position can be arbitrary and the internal resistance is zero (ideal source). The value of force/torque instead is equivalent to the present value of the selected time function at each calculation step.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [N]/[Nm]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin1-Pin2 Force Source	TRB1-TRB2/TR1-TR2	translational/ translational_v
Pin1-Pin2 Torque Source	ROTB1-ROTB2/ROT1-ROT2	rotational/ rotational_v

Spring

>>Basics>Physical Domains>Mechanical>Displacement-Force-Representation

>>Basics>Physical Domains>Mechanical>Velocity-Force-Representation

The component represents a simple spring specified by a spring rate.

$$F(t) = C \cdot s(t)$$

$$M(t) = C \cdot \varphi(t)$$

Dialog Settings	
⭳⭱	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. **«Name»«Default»«Unit»«Description»** Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Spring Rate [N/m]/Spring Rate [Nm/rad]	C/C	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin1-Pin2 Translational Spring	TRB1-TRB2/TR1-TR2	translational/ translational_v
Pin1-Pin2 Rotational Spring	ROTB1-ROTB2/ROT1-ROT2	rotational/ rotational_v

Damper

>>Basics>Physical Domains>Mechanical>Displacement-Force-Representation

>>Basics>Physical Domains>Mechanical>Velocity-Force-Representation

The component represents a simple damper specified by a linear damping coefficient.

$$F(t) = DAMPING \cdot v(t)$$

$$M(t) = DAMPING \cdot \omega(t)$$

Dialog Settings	
⭳⭱	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. **«Name»«Default»«Unit»«Description»** Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Damping Coefficient [N*s/m]/Damping Coefficient [Nm*s/rad]	DAMPING/DAMPING	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin1-Pin2 Translational Damper	TRB1-TRB2/TR1-TR2	translational/ translational_v
Pin1-Pin2 Rotational Damper	ROTB1-ROTB2/ROT1-ROT2	rotational/ rotational_v

Mass

>>Basics>Physical Domains>Mechanical>Displacement-Force-Representation

>>Basics>Physical Domains>Mechanical>Velocity-Force-Representation

The component represents a mass specified by a mass/inertia and initial position/velocity.

$$F(t) = M \cdot a(t) \qquad s(t) = \int v\,dt + s_0 \qquad v(t) = \int a\,dt + v_0$$

$$M(t) = J \cdot a(t) \qquad \varphi(t) = \int \omega\,dt + \varphi_0 \qquad \omega(t) = \int a\,dt + \omega_0$$

Dialog Settings	
↓↑	**«Value»** Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. **«Name»«Default»«Unit»«Description»** Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Mass [kg]/Mass Moment of Inertia [kg*m^2]	M/J	real
Initial Position [m]/Initial Angle [rad]	S0/PHI0	real
Initial Velocity [m/s]/Initial Angular Velocity [rad/s]	V0/OMEGA0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real
Acceleration [m/s²]/Acceleration [rad/s²]	ACC/ACC	real

Component Nodes

Description	Node Name	Nature
Pin1-Pin2 Translational Mass	TRB1-TRB2/TR1-TR2	translational/ translational_v
Pin1-Pin2 Rotational Mass	ROTB1-ROTB2/ROT1-ROT2	rotational/ rotational_v

Spacer

>>Basics>Physical Domains>Mechanical>Displacement-Force-Representation

The component represents a constant displacement between two components in a mechanical system specified by a length.

$$v(t) = \frac{ds}{dt} \qquad s = LEN$$

$$\omega(t) = \frac{d\varphi}{dt} \qquad \varphi = LEN$$

Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. **«Name»«Default»«Unit»«Description»** Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Length [m]/Length [rad]	LEN/LEN	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin1-Pin2 Translational Spacer	TRB1-TRB2	translational
Pin1-Pin2 Rotational Spacer	ROTB1-ROTB2	rotational

Limit Stop

>>Basics>Physical Domains>Mechanical>Displacement-Force-Representation

>>Basics>Physical Domains>Mechanical>Velocity-Force-Representation

The component represents a parallel damper and spring combination within defined limits (upper and lower limit).

$$\begin{aligned}&\text{if}\quad s(t) > SUL\\&F(t) = C\cdot(s(t)-SUL)+DAMPING\cdot v(t)\\&\text{if}\quad \overline{s(t) < SLL}\\&F(t) = C\cdot(s(t)-SLL)+DAMPING\cdot v(t)\\&\text{else}\quad F(t) = 0\end{aligned}$$

$$\begin{aligned}&\text{if}\quad \varphi(t) > PHIUL\\&M(t) = C\cdot(\varphi(t)-PHIUL)+DAMPING\cdot\omega(t)\\&\text{if}\quad \overline{\varphi(t) < PHILL}\\&M(t) = C\cdot(\varphi(t)-PHILL)+DAMPING\cdot v(t)\\&\text{else}\quad M(t) = 0\end{aligned}$$

Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Spring Rate [N/m]/Spring Rate [Nm/rad]	C/C	real
Damping Coefficient [N*s/m]/Damping Coefficient [Nm*s/rad]	DAMPING/DAMPING	real
Upper Position Limit [m]/Upper Angle Limit [rad]	SUL/PHIUL	real
Lower Position Limit [m]/Lower Angle Limit [rad]	SLL/PHILL	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Displacement [m]/Angle [rad]	S/PHI	real
Force [N]/Torque [Nm]	F/M	real
Velocity [m/s]/Angular Velocity [rad/s]	V/OMEGA	real

Component Nodes

Description	Node Name	Nature
Pin1-Pin2 Translational Limit	TRB1-TRB2/TR1-TR2	translational/ translational_v
Pin1-Pin2 Rotational Limit	ROTB1-ROTB2/ROT1-ROT2	rotational/ rotational_v

10.4 Thermal Components

- Temperature Source (T)
- Heat Flow Source (H)
- Thermal Resistance (RTH)
- Thermal Capacitance (CTH)

Across and Through Quantity of the Thermal Domain

Across	Through	Equation
Temperature [K]	Heat Flow [J/s]	$T(t) = K \cdot H(t)$

The thermal domain components appear in the *Thermal* folder of *Physical Domains* in the «Basics» tab. The folder provides thermal basic components to model simple thermal equivalent circuits, for example cooling systems or heat sinks. The thermal domain uses the temperature as across and the heat flow as through quantity.

By default, components and wires of the thermal domain are represented in red in the graphical representation of the Schematic. Select SHEET>PROPERTIES to change the wire color of the domain. You can only change the wire color. The symbol color is not affected by the setting.

Temperature Source

>>Basics>Physical Domains>Thermal

The component represents an ideal temperature source which supplies the defined temperature for an arbitrary heat flow. Within the dialog you can choose a constant or the time-controlled type.

Dialog Settings	
⊙	«Temperature [K]» The linear temperature source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⊙	«Time Controlled» The time-controlled temperature source is selected. In the list, select a time function and define the corresponding parameters. See also "Using Time Functions" on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Temperature Source

$T(t) = VALUE$

The temperature is equivalent to the present value of the quantity specified in the text box. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The heat flow through an ideal temperature source can be arbitrary. Real temperature sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Temperature [K]	VALUE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Temperature [K]	T	real
Heat Flow [J/s]	H	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	TH1	thermal

10 Time-Controlled Temperature Source

$T(t) = f[\text{time function}]$

The time controlled temperature sources have the same properties as the linear temperature source above; the heat flow can be arbitrary and the internal resistance is zero (ideal source). The value of the temperature differential instead is equivalent to the present value of the selected time function at each calculation step.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [T]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Temperature [K]	T	real
Heat Flow [J/s]	H	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	TH1	thermal

Heat Flow Source

>>Basics>Physical Domains>Thermal

The component represents an ideal flow source which supplies the defined flow rate for an arbitrary pressure differential. Within the dialog you can choose a constant or the time-controlled type.

Dialog Settings	
⊙	«Heat Flow [J/s]» The linear heat flow source is selected. Common parameter type. Enter a numerical value, a variable, or an expression in the text box or use the pin to connect a quantity.
☑ ☐	«Use pin» If the box is checked, the value is defined by the quantity connected to the pin, otherwise the value in the text box is used.
⊙	«Time Controlled» The time-controlled heat flow source is selected. In the list, select a time function and define the corresponding parameters. See also “Using Time Functions” on page 275.
▼	In the list, select the sine, pulse, triangular, trapezoidal, saw-tooth rising, or saw-tooth falling time function.

Linear Heat Flow Source

$$H(t) = VALUE$$

The flow rate is equivalent to the present value of the quantity specified in the input field. Usually a constant value, a time function, or a variable are chosen, or the value is determined in a state graph. All system quantities and variables can be combined with mathematical and logical operators and constants.

The pressure of an ideal flow source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Heat Flow [J/s]	VALUE	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Temperature [K]	T	real
Heat Flow [J/s]	H	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	TH1	thermal
Pin 2	TH2	thermal

Time-Controlled Heat Flow Source

$$H(t) = f[\text{time function}]$$

The time controlled heat flow sources have the same properties as the linear heat flow source above; the temperature can be arbitrary and the internal resistance is zero (ideal source). The value of the flow rate instead is equivalent to the present value of the selected time function at each calculation step.

The simulator internal names of time controlled flow sources read as follows: Sine – HSINE; Pulse – HPULSE; Triangular – HTRIANG; Trapezoidal – HTRAPEZ; Saw-tooth Rising – HSZ_RISE; Saw-tooth Falling – HST_FALL. The names are only relevant to simulation scripts in SML. Access to source parameters is carried out only by the component name.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Amplitude [J/s]	AMPL	real
Frequency [Hz] (no trapezoidal function parameter)	FREQU	real
Period [s]	TPERIO	real
Delay [s]	TDELAY	real
Phase [deg]	PHASE	real
Angular Dimension	n/a	
Offset	OFF	real
Periodical; PERIO=1, periodical function, PERIO=0; non-periodical function	PERIO	real
Damping factor [1/s] (only sine function) DAMPING>0, amplitude is exponentially decreasing (damped). DAMPING<0, amplitude is exponentially increasing, DAMPING=0, no damping	DAMPING	real
Rise Time [s] (only trapezoidal function)	TRISE	real
Fall Time [s] (only trapezoidal function)	TFALL	real
Pulse Width [s] (only trapezoidal function)	PWIDTH	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Temperature [K]	T	real
Heat Flow [J/s]	H	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	TH1	thermal
Pin 2	TH2	thermal

Thermal Resistance

>>Basics>Physical Domains>Thermal

The component represents a ideal thermal resistance specified ba a linear coefficient.

$$T(t) = K \cdot H(t)$$

Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below. «Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Linear Coefficient [K*s/J]	K	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Temperature [K]	T	real
Heat Flow [J/s]	H	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	TH1	thermal
Pin 2	TH2	thermal

Thermal Capacitance

>>Basics>Physical Domains>Thermal

The component represents a ideal thermal capacitance.

$$H(t) = C_{TH} \cdot H(t) + T0$$

Dialog Settings	
↓↑	«Value» Click the **«Value»** field of the corresponding parameter to define a value. Common parameter types. Enter a numerical value, a variable, or an expression. Look the parameter meaning up in the parameter table below.
	«Name»«Default»«Unit»«Description» Displays the parameter names and their corresponding unit, default value, and description. You cannot make changes in these fields. If you want to add user-specific information to a parameter, use the «Info» field in the component output dialog.

Component Parameters

Description [Unit]	Parameter Name	Data Type
Thermal Capacitance [J/K]	C_TH	real
Initial Temperature [K]	T0	real

Component Outputs

Description [Unit]	Parameter Name	Data Type
Temperature [K]	T	real
Heat Flow [J/s]	H	real

Component Nodes

Description	Node Name	Nature
Pin 1 (with red point)	TH1	thermal

11 Schematic Environment

<u>This chapter contains information on:</u>

- Managing files in Schematic
- Printing files
- Updating models from former versions
- Using templates
- Modifying representations and displays in the model sheet
- Arranging Schematic sheets
- Exchanging data with other applications

11.1 File Management in Schematic

The file for a simulation model created in Schematic has the file extension .ssh. The graphical model description, simulation parameters, and output settings are saved in the model sheet.

If you use external data for characteristics (.mdx, .cls, .csv files), make certain that these files are available when you relocate the .ssh file. All other files (.sml, .brs, .prc, .smt), which have the same root file name as the model sheet, are regenerated when you start a new simulation run.

Using SIMPLORER's project management feature, you can easily copy and move complete project data, provided that a project is used for a simulation task. See also 2.5 "Project Management with the SSC Commander" on page 14.

Using Files in the Schematic

In Schematic, you can load, save, and print files from the FILE menu or from toolbar symbols. Files created or modified within Schematic can be included in the active project. The files can then be loaded directly when you open Schematic from the SSC Commander.

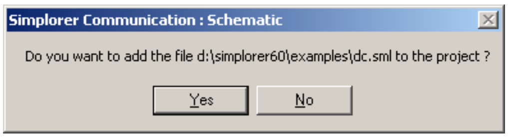
File Menu ▾

Command	Toolbar Symbol and Description
NEW	Opens a new window to create a new Schematic file.
OPEN	Opens an existing file. The file list can be managed using file filters. A preview is displayed on the right side of the dialog.
READ BRS FILE	Writes parameter names and values from an external parameter file (.brs) into the model sheet.
CLOSE	Closes the active document and prompts you to save it if changes were made to the last saved version.
CLOSE ALL	Closes all open documents and prompts you to save them if changes were made to each last saved version.
SAVE	Saves the active document and prompt for the file name if the document is being saved for the first time.
SAVE AS	Similar to SAVE, but the file name is required every time.

Command	Toolbar Symbol and Description
SAVE ALL	Saves all documents open in the Text Editor, prompting for a file name when necessary.
PROJECT	Switches to the SSC Commander window.
IMPORT	Inserts the elements of a Schematic file into the active file.
EXPORT	Exports the selected elements into a new Schematic file. There are two supported version formats: .ssh file 6.0 (standard) and .ssh file 5.0.
MODEL DESCRIPTION	Creates a separate .sml or .vhd file from the active simulation model (without starting a simulation). See also "Model Description" on page 408.
PRINT	Prints the active Schematic sheet. Defines the print pages, the number of copies, and the destination (printer or file).
PRINT PREVIEW	Displays a full-page view of the active Sheet. The print layout is defined in SHEET>PAGE SIZE. See also "Printing Model Sheets" on page 341.
PRINTER SETUP	Defines the print options: paper size, paper feed, and format (portrait or landscape).
SEND WITH E-MAIL	Creates a new e-mail message, and attaches the active Schematic sheet file. You can also set the option to zip the file and include the .sdb file belonging to it. See also "Sending Simulation Models in E-mails" on page 341.
«FILE LIST»	List of the last 4 opened files.
EXIT	Closes Schematic and prompts you to save any open documents.

Project Files

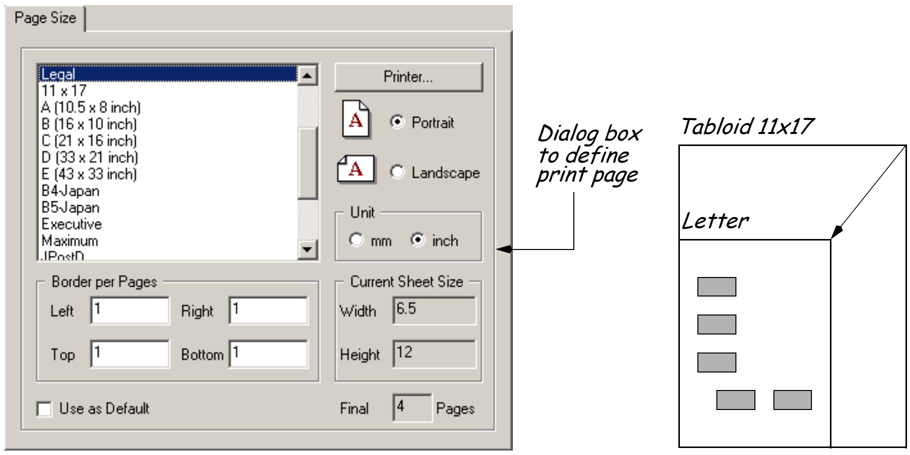
The below dialog appears box if you have not included the document in the project, or if you are saving it for the first time. If you click <No>, the dialog box is displayed again if you save the file under the same name.



Dialog box to add a file to the active project

Printing Model Sheets

Use the SHEET>PAGE SIZE menu to define sheet size and format for model sheets. When you reduce the sheet size, the objects on the sheet must be placed on the lower left side, to avoid an error indicating objects are outside of the sheet.



To print a model sheet, choose FILE>PRINT, or click the print symbol on the toolbar. If you select print options, you can decide whether the model sheet should be fitted to the page or printed in its original size. If you do not select the **«Fit to page»** option, then the number of pages printed depends on the model sheet and page size.

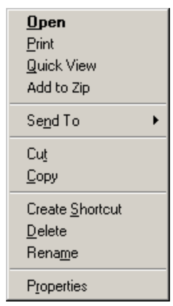
Only visible levels are printed. See also 11.4 “Arranging Screen Layout and Using Help” on page 360.

Quick View Display

The Quick View command displays and prints Schematic files directly from the Windows Explorer. The **«Quick View»** command on the shortcut menu opens the Schematic file in the view mode. You can print the file with the FILE>PRINT menu command.

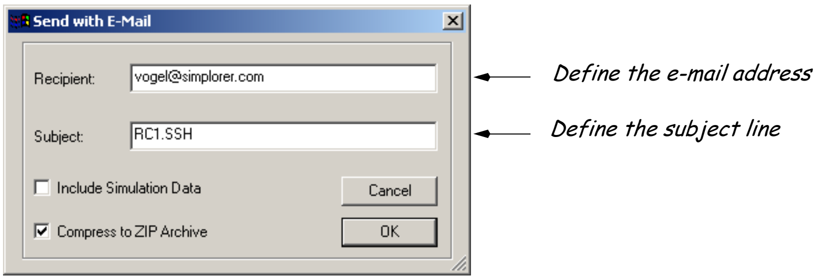
The quick view display merely provides a brief overview of the file contents; its resolution is not as high as the resolution available in Schematic itself.

Shortcut menu of a Schematic file in the Windows explorer with Quick View function



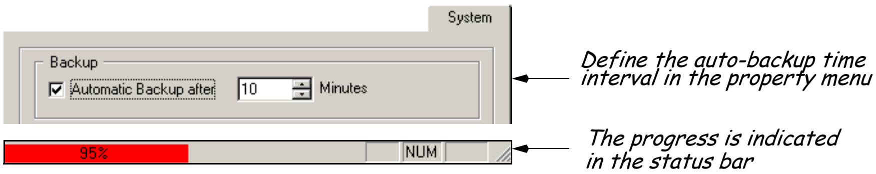
Sending Simulation Models in E-mails

The FILE>SEND WITH E-MAIL command opens a dialog to send the active model file as e-mail. Define an e-mail address and a subject line. By default, the subject line is the name of the model sheet. The option **«Include Simulation Data»** adds an existing result file (model name.sdb) to the e-mail. The option **«Compress to ZIP Archive»** creates a .zip files with the selected files. You do not need an own ZIP program on your PC because SIMPLORER uses an internal program to zip files. The .zip file generate by SIMPLORER can be opened with a WinZip[1] application.



Auto-Backup Function (Time Interval)

The backup time interval for saving open model sheets in the Schematic is user-defined and can also be switched off if desired. After a system errors happens you can decide if you want to use the backup file, ignore, or delete them. The **«Backup»** option is in the **«System»** tab of the sheet properties dialog (open with SHEET>PROPERTIES).



Updating Model Sheets from Former Versions

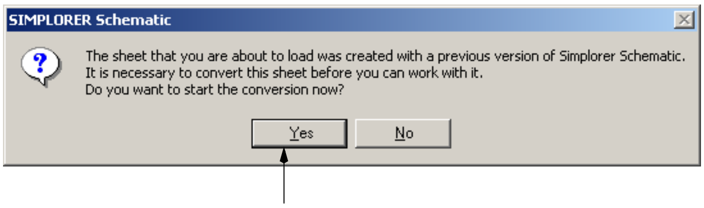
Models from former SIMPLORER versions use components that do not support all functions of SIMPLORER version 6.0. To replace these components with components from new libraries, you must update the sheet. You can perform updates in the following situations:

- As a file is opened (conversion 4.x into 6.0 or synchronize 5.0 files)
- With the SHEET>SYNCHRONIZE command
- When updating separate components via the **«Synchronize»** command on an element's shortcut menu

Updating does not change the original file but creates a **new** Schematic file you need to save under a new name.

All components used in a former sheet must be available in an **installed** model library of the new SIMPLORER 6.0 format. You must convert model libraries (for 4.x sheets) or synchronize them (for 5.0 sheets) before you can update the sheet.

1. Nico Mak Computing, Inc. and Top Systems GmbH 1991-1997



SIMPLORER prompt to convert a 4.x sheet

To synchronize or convert libraries, start the FILE>INSTALL LIBRARY command in the Model Agent and enter a name for the new library if the prompt to save appears.

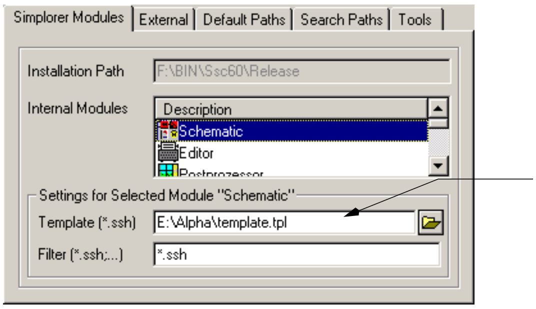
When during the conversion process an ambiguous component definition occurs (the component is contained in more than one installed library), an additional dialog is opened. Select the component from the library you want and continue the process. Components with no equivalent model in SIMPLORER 6.0 format is deactivated and hatched after loading the sheet. See also “Converting 4.x Libraries” on page 485.

Using Templates

In the SSC Commander you can decide which file (if any) you want to use as a template for new Schematic models.

Creating and Using Templates in the Schematic

1. Create a Schematic file with default elements for each new model sheet (drawing elements, text elements with placeholders, Schematic settings and so on).
2. Save this file with the extension .ssh using FILE>SAVE AS or FILE>SAVE.
3. In the SSC Commander, select OPTIONS>PROGRAM DIRECTORIES to define the name and path for the template file.



Path and name of template file

The elements of the template file are included in each new model file created with the template. If you do not want to use a template file, omit the entries for path and name.

Modifying the Model Bar

You can specify which components are available on the Schematic model bar. Each library model can be inserted. To place a model bar component on the sheet, click the symbol and release the mouse button. The component “sticks to” the mouse pointer. Click in the sheet to define the component position.

Creating and Using a Model Bar in the Schematic

1 Decide which components to include on the model bar.

2 Create a symbol (.bmp file with 16x15 pixel and 16 colors) for each of component you want to include.

3 Enter the symbol in the corresponding property dialog for each component:

a. Click the **«Files»** tab of the component property dialog (double-click the name in the Model Agent).

b. Click <Add> to insert a file. In the box above, you can define a file for the selected lanuage in the dialog, in the box below, for all languages .

c. Select the **«Model Bar»** file type and check the **«Intern»** box to include the file in the library.

d. Click <OK>.

4 Modify the *POWDER.INI* file (Database and component ID) as follows:

a. Open your user *POWDER.INI* in your Windows system folder.

b. Enter the following entries after the Model Bar section.

c. Save the file.

INI Text	Description
[ModelBar]	Start of the model bar section
NumberOfModels=5	Number of models including separators in the bar.
Database0=E:\alpha\macro.smd ModelID0=1	First entry with index 0 and library path. Model ID in the defined library.
Database8=.\DISPLAY.SMD ModelID8=2	Second entry with the index 1 and library path. Model ID in the defined library.
Database4=.\Basic Elements.smd ModelID4=32	Third entry with the index 4 and library path. The missing index 3 creates a separator before the symbol.
Database5=.\Basic Elements.smd ModelID5=30	Fourth entry with the index 5 and library path. Model ID in the defined library.

The model ID of a library component is displayed in the **«Library»** dialog of the dropped component on the sheet. The *POWDER.INI* is located in the Windows system folder when Schematic is opened, otherwise the file is in the corresponding SIMPLORER folder with user settings. After changing the .ini file you must (close) start Schematic.

11.2 Modifying the Representation and Displays of the Model Sheet

In addition to the simulation model itself you can add drawing elements and text to document the model. With special text macros system information as date and document name can be inserted directly on the sheet. All graphical elements and component displays have their own property dialog where colors, line, and fill styles are defined.

Drawing elements and component displays can be aligned and moved independent of the component grid. The ELEMENT>MOVE UP and MOVE DOWN commands move drawing elements into the foreground or background. So you can change the sequence of them later.

Most of the property dialogs have a **«Use as Default»** option. If the box is checked, all settings of this dialog are applied for all new identical element types placed on the sheet. If the option **«For All»** is selected, the active settings of the dialog are applied for all identical element types on the sheet.

11.2.1 Using Standard Edit Commands

Through commands in the EDIT menu, symbols in the toolbar, or hot keys, all components and elements of the model sheet can be deleted or copied and inserted at a new location on the sheet. These commands always apply to selected elements.

Selecting Elements

Elements (SIMPLORER components, connections, names, graphic objects, text) are selected by clicking them with the mouse. A selected element is highlighted with a red outline. Several elements can be selected simultaneously by pressing the SHIFT key. To select an unfilled element you must click the ouline of the element.

All elements in an area can be selected by drawing a rectangle around the area. Place the mouse pointer at any corner of the area you want to select. Press and hold the mouse button and drag the cursor to the opposite corner. If you press, in addition, the SHIFT key, all of the rectangle touched elements are add to the selection. Otherwise, only the complete enclosed elements are selected.

Remove the selection by clicking anywhere on the sheet outside the selection.

Moving Elements

Place the mouse pointer in the highlighted area, and drag the selected elements with pressed mouse button to a new position. All relations between the elements are kept.

Moving Elements with Arrow Keys

Selected sheet elements can be moved step-by-step to the left, right, up, or down using the arrow keys. If you press, in addition, the CTRL key, the visible area is scrolled in the window.

Copying, Cutting, Deleting, and Inserting Elements

Selected elements can be copied with CTRL+C, deleted with CTRL+X, and removed with the DEL key. When a component has been copied or deleted, it can be inserted with Ctrl+V in the same or another sheet.

The commands CUT and COPY refer to the active selection. If nothing is selected on the sheet, these commands are not available.

You can also insert copied data into the clipboard in other Windows applications. See also 11.3 “Data Exchange with Other Applications” on page 358.

Edit Menu

Command	Toolbar Symbol and Description
UNDO	Cancels or reverses the last input or action.
REDO	Restores an input or action.
CUT	Removes the active selection and places it into the clipboard.
COPY	Copies the selected elements into the clipboard.
OLE COPY	Copies the selected elements as an OLE object into the clipboard. (Object that supports linking and embedding.)
PASTE	Pastes data from the clipboard.
DELETE	Deletes selected elements.
PASTE SPECIAL	Inserts clipboard contents as a bitmap.
DUPLICATE	Copies and inserts the selected element. All subsheets created with duplicate will always have the same content, because any changes are updated in all corresponding subsheets.
COPY DATA FROM	Copies data from one element to another.
SELECT ALL	Selects all elements on the sheet.
SEARCH/REPLACE	Searches and replaces character strings in text and model parameters.
INSERT NEW OBJECT	Pastes an existing object or opens an application to create an object and inserts the object into the model sheet.
LINKS	Displays all links in the model sheet with other windows applications.
OBJECT	Displays properties of embedded objects and opens a dialog to convert the objects and to start the original application.
PROPERTIES	Opens the property menu of the selected object.

11.2.2 Displaying Names and Parameters of Components and Connections

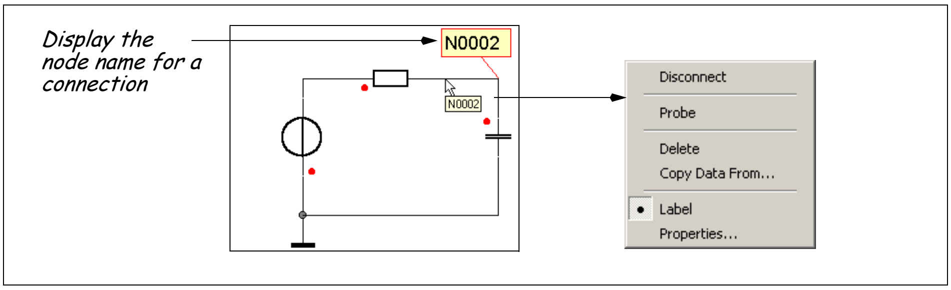
Each component is defined by its name and its parameters. Schematic attaches and displays these specifications directly on the sheet. There are many options for the representation of names and parameters. Whether all the information can be displayed depends on the number of components and the size of the sheet. You may modify a sheet according your personal preferences.

The following properties can be displayed:

- Pin names
- Component names
- Parameter values
- Node names

Displaying Node Names

For a selected connection the Properties menu can be opened with the right mouse button. The «Label» option sets the display of the node names. If you move the label on the sheet, the connection line goes to the next point at the wire.



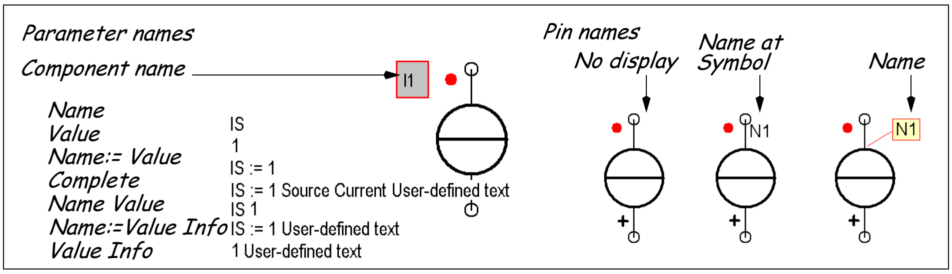
Defining Displays for Parameter and Pin Names

Within the «Output/Display» dialog in the property menu of the components (blocks, states), the format of the display can be supplemented. The «Show» column contains the settings for graphical output. The drop down menu with the items is opened with a mouse click the «Show» field. The meaning of the options is shown in the figure above.

If you cannot see the «Show» column, reduce the size of the first column by moving the sliders to the left.

The «Complete» option means that all available outputs are displayed, as well as the information field.

If the parameter pin is available on the sheet, the linking line of a defined display point at the corresponding pin.

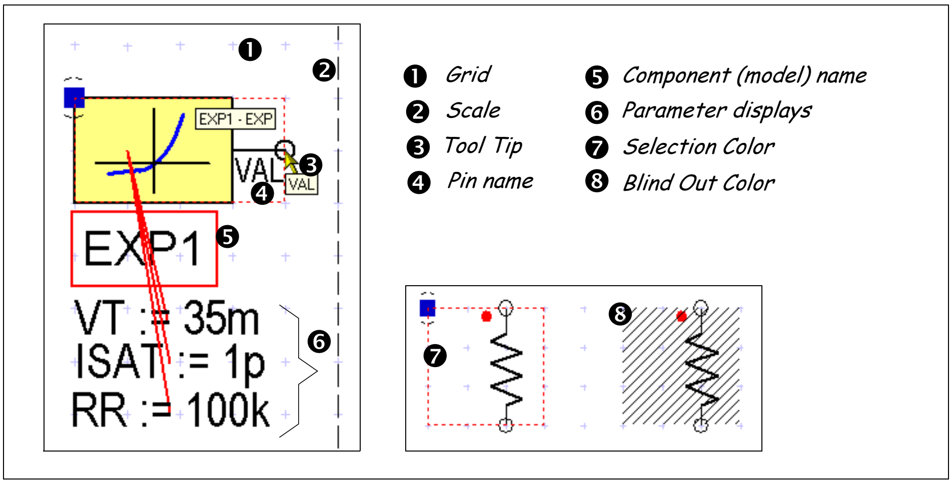


Setting the Display for all Components

The options in SHEET>PROPERTIES **«Sheet»** sets the sheet properties itself as well as the displays for names and parameters of components. For parameter and pin names you can select **«Show Never»**, **«Show as in Library»**, or **«Show Always»**. The settings are applied to just dropped components, not to components which are already on sheet.

In addition, you can change the settings of nature colors. Please note: You can only change the wiring color, not the color of component symbols.

The option **«Use as Default»** applies the settings defined in the dialog to all new model sheets.



Sheet Menu ▾

Command	Description
DETERMINE BLOCK SEQUENCE	Determines the signal direction in block diagrams. See also "Modeling with Block Diagrams" on page 201.
PAGE SIZE	Defines print page size. See also 11.1 "File Management in Schematic" on page 339.

Command	Description
DISPLAY	Defines settings for the display of a model sheet. See also 11.2 “Modifying the Representation and Displays of the Model Sheet” on page 345.
SYNCHRONIZE	Updates components in the model sheet. See also “Updating Model Sheets from Former Versions” on page 342.
SUBSHEET	Creates and edits subsheets. See also 3.5 “Subsheets in Simulation Models” on page 57.
PROPERTIES	Dialog window with simulation parameters and settings for Schematic displays.

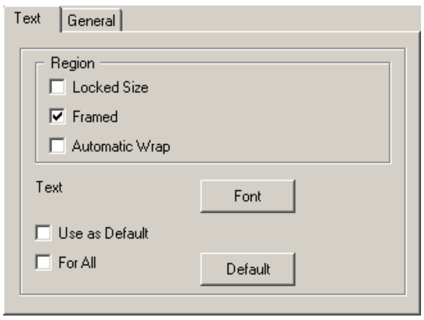
Changing Properties of Parameter Displays and Text Elements

Each name element on the sheet has a property menu which can be opened with a right mouse click. The **«Text»** and **«General»** tabs contain options for color, font, and frame.

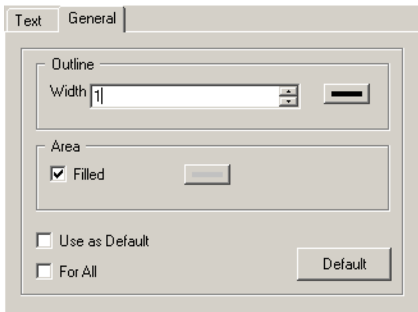
You can use the settings for:

- The selected element
- All name elements on the sheet
- As default

depending on whether you select the corresponding check box.



Text options for parameter displays

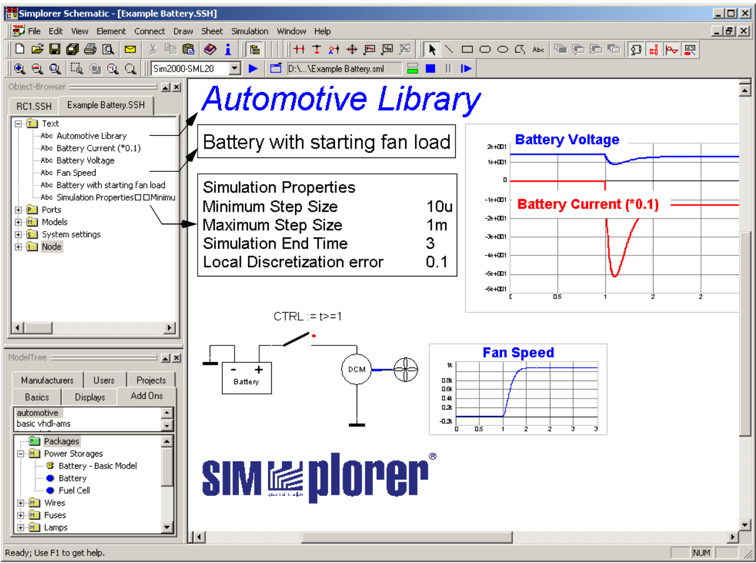


Graphical Settings for parameter displays

Double-click the corresponding color control box. A dialog for color selection appears. If the option **«Use as Default»** is selected, the settings are valid for all following components. The option **«For All»** applies the settings to all components on the active sheet.

11.2.3 Using the Text Tool

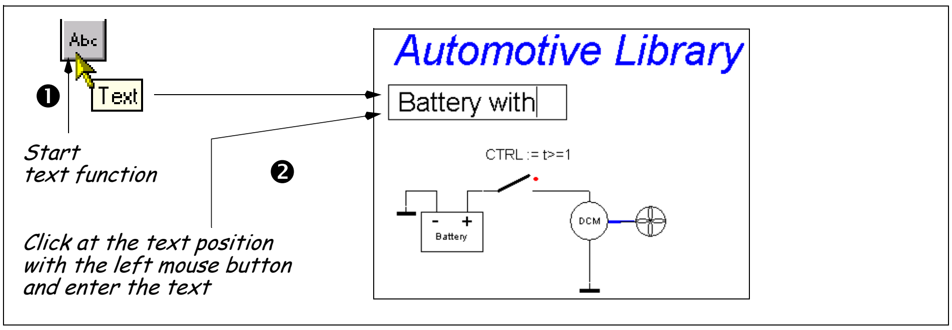
With the text tool you can insert text elements at each position on the sheet. The text element has a property dialog, with options for font, color, frame, and other settings.



Text elements in a model sheet

Activating Text Mode and Defining Text Position

To create text elements in the model sheet, start the text mode, either with DRAW/TEXT or with the text symbol in the toolbar. The mouse pointer becomes a cursor; click with left mouse button at the desired insertion point and enter the text. Click outside the text or press ESC to close the text mode.



Inserting Date, Time, and File Information

You can also insert into each text element specific information about the model sheet, such as the time and the date. This information is updated automatically as it changes.

You can insert the following information:

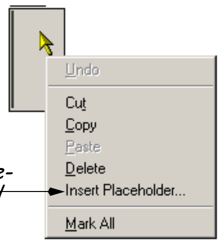
- Date and time
- Parameters of models
- Name of model file and name of project file
- System settings (simulation parameters, trigger times, outputs)

Inserting New Text with Placeholders

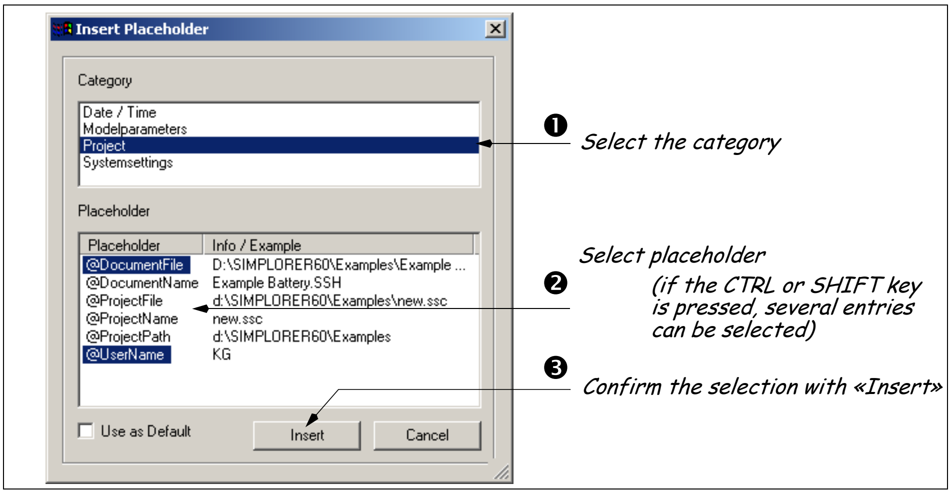
1 Start the text mode (DRAW>TEXT or text symbol Abc) and click with the **left** mouse button at the desired insertion point in the sheet.

2 Click with the **right** mouse button the text field and select **«Insert Placeholder»**.

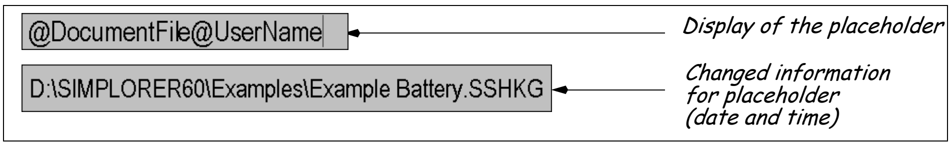
3 Select an entry in the **«Category»** window, the type of the placeholder in the **«Placeholder»** window and confirm with <Insert>.



To select more than one entry in the list press the CTRL or SHIFT key and click all entries you want. All entries are inserted with line break in the text element.

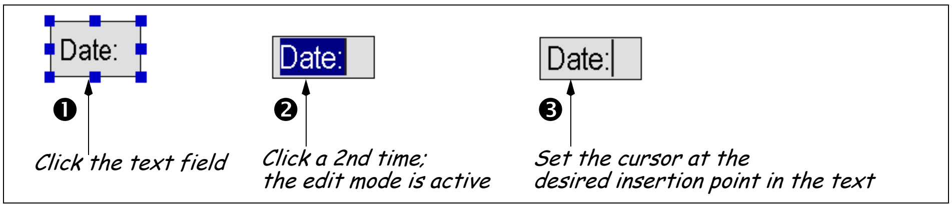


4 Click **outside** the text field in the sheet or press ESC; the size of the display of the placeholder is automatically adapted to the corresponding information.



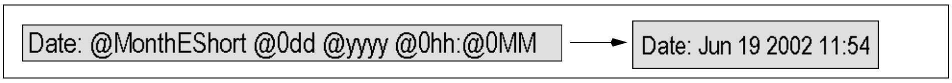
Extending an Existing Placeholder with Text

1 Click the text field on the sheet two times, but do NOT double-click. Edit mode starts.



2 Click with the right mouse button in the text field, and choose **«Insert placeholder»** from the shortcut menu.

3 Select the category and the placeholder type, and click <Insert> to apply the text element selection. Click outside the text field or press ESC to close the text mode.



The text is inserted at the cursor position. The notation of other placeholders can thus be changed to display other information.

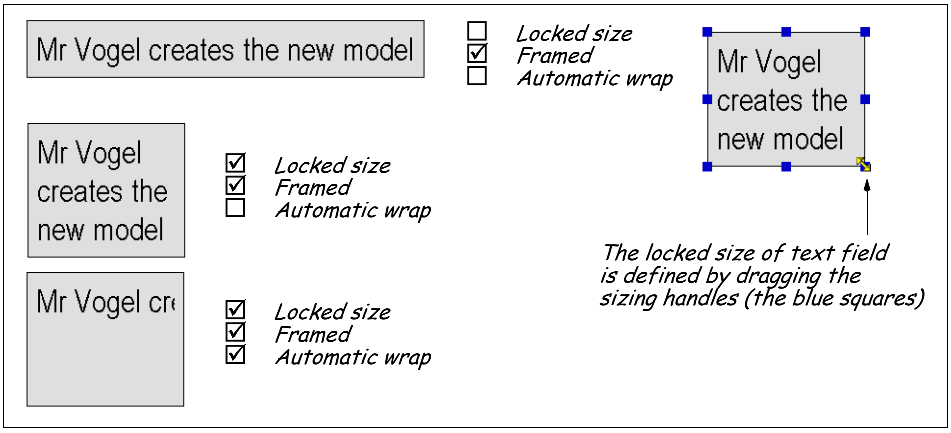
Changing Predefined Placeholders

You can change the notation of placeholders in the text and extend them with other function characters. Placeholders always begin with the '@' sign. The following characters are interpreted as functions.

To display, e.g., `TEND`, `HMIN`, and `HMAX` on the Sheet, you can insert a text element with the following string: *HMIN: @TR.HMIN, HMAX: @TR.HMAX, TEND: @TR.TEND*. The display on the sheet appear in this form: HMIN: 5u, HMAX: 1m, TEND: 15m. If you copy this text element and insert it into a different sheet, it will display the active values for these simulation parameters for the sheet where it has been inserted.

Size and Form of Text Elements

The size of the text element either automatically adapts to the text inside or remains a fixed size independent of the number of characters in the text field. The adaptation depends on the settings in the property menu of the text element. Double-click the element or choose EDIT>PROPERTIES to open the property dialog.

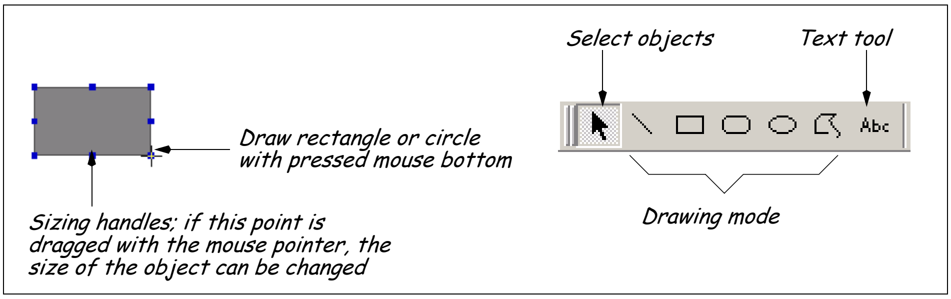


11.2.4 Drawing Elements

In addition to displays of component names and parameters, Schematic provides several drawing elements to enhance the model sheet presentation.

Inserting Drawing Elements

These drawing elements are created in draw mode, which you can start with the DRAW menu or the Draw toolbar. Once in draw mode, the mouse pointer becomes a cross wire, and you can draw lines, rectangles, rounded rectangles, ellipses, or polygons on the sheet. Click on the sheet and draw the element with pressed mouse button. If you press, in addition, the SHIFT key, you can create a quadrate in the rectangle and a circle in the ellipse mode. To finish draw mode, press the ESC key or double-click the sheet background.



Draw Menu ▾

Command	Toolbar Symbol and Description
SELECT	Starts select mode (to alter or move shapes, lines, or text blocks).
LINE	Draws a line.

Command	Toolbar Symbol and Description
RECTANGLE	Draws a rectangle.
ROUNDED RECTANGLE	Draws a rectangle with rounded corners.
ELLIPSE	Draws an ellipse.
POLYGON	Draws a polygon.
TEXT	Abc Starts the text mode.
DATE / TIME	Inserts a Text element with active time and date. Wed Jun 19 2002 12:06
PROJECT INFORMATION	Starts text mode and opens the dialog for inserting project parameters.

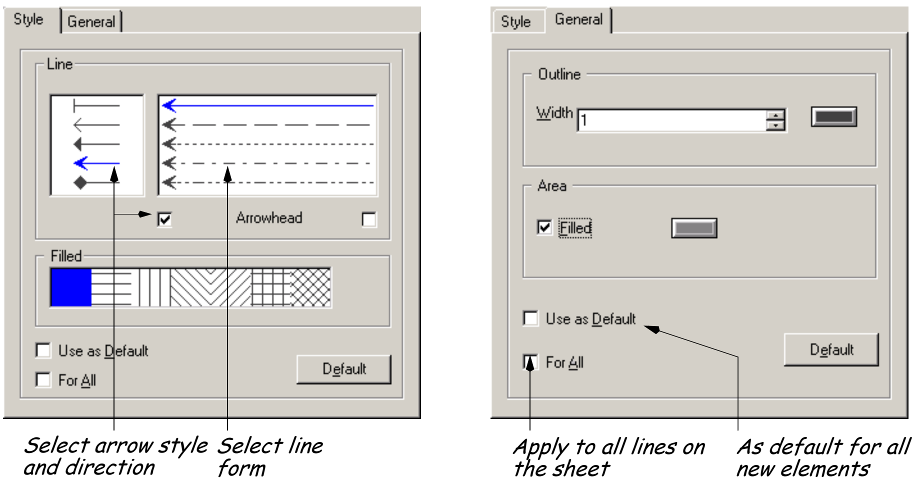
Properties of Drawing Elements

Each drawing element has a property dialog with the format settings. Double-click the element or choose EDIT>PROPERTIES to open this dialog.

Properties of a Line

The following settings are possible for lines:

- Format (continued, pointed, dashed, . . .)
- Arrow (at beginning, at the end or both)
- Width
- Color

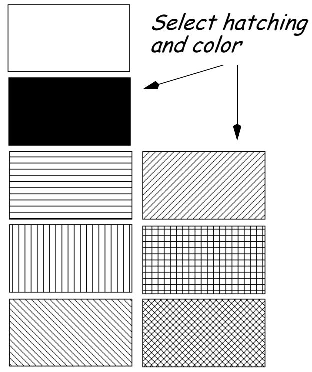


Properties of Rectangles, Circles, and Rounded Rectangles

In addition to the properties of the outline, you can define the hatching and color of areas in rectangles, circles, and rounded rectangles.

The following settings are available:

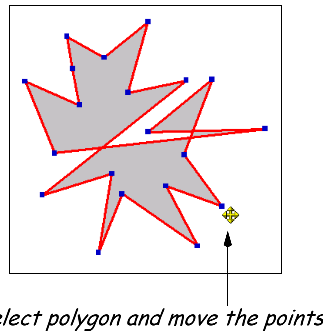
- Area as color sheet or hatching (EDIT>PROPERTIES **«Style»**)
- Color of the area (EDIT>PROPERTIES **«General»**)
- Outline or area (with or without fill in area in EDIT>PROPERTIES **«General»**)



Properties of Polygons

The corners of the polygon are defined with a mouse click. A double-click closes the drawing mode. You can change the form of the selected polygon later by moving the corners with the mouse pointer to a new location (the mouse pointer becomes a cross).

For a polygon, you can define the outline and the color for the area.



11.2.5 Arranging and Aligning Displays, Text, and Drawing Elements

Component displays, text, and drawing elements can be aligned and moved independent of the component grid. The ELEMENT>MOVE UP and MOVE DOWN commands move drawing elements into the foreground or background. So you can change the sequence of them later.

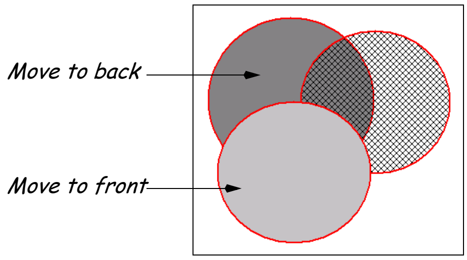
To change the size of drawing elements, drag on the element handels with the mouse. If you press, in addition, the SHIFT key, you can adjusting the size proportionally.

Moving Elements

You can select each element on the sheet. When selected, an element is highlighted with a red outline and the blue selection points become visible. You can move the element to a new location by pressing and holding the left mouse button. To select unfilled elements you must click the element outline.

Arranging Elements

Elements may be hidden behind other elements on the sheet. Depending on the arrangement, the elements in the upper level may hide the elements in the lower level. You can change the sequence of arrangement with the ELEMENT>MOVE UP / MOVE DOWN menu commands. These commands are always applied to the selected element.

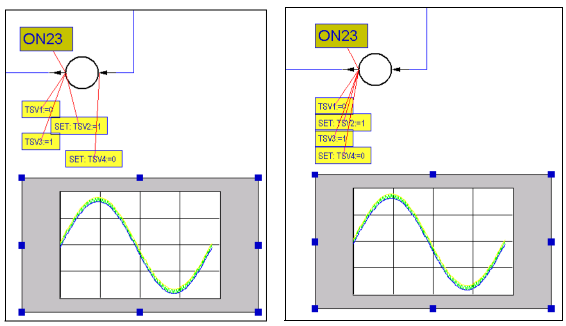


Aligning Elements

If several elements are selected at the same time, you can align these elements with their selection points, either with the commands in the ELEMENT>ALIGN menu or the commands in the standard toolbar.

The alignment is always oriented to the **last** selected element.

Aligning the Marked Elements on the Left Side



Setting the Same Width and Same Height for Elements

These commands adapt several selected drawing text and display elements to the same height and width. The size for width and height is always oriented to the **last** selected element. You can start these functions with the commands in the ELEMENT>SIZE menu or with the commands in the standard toolbar.

Element Menu

Command	Toolbar Symbol and Description
OUTPUT	Changes the outputs of a selected element (only for components).
PARAMETER	Changes the parameters of a selected element (only for components).
DON'T ADD TO SML	Deactivates the selected component or components. Components thus flagged are ignored by the simulator.
LINE COLOR	Changes the line color of a selected drawing element.
FILL COLOR	Change the fill color of a selected drawing element.
FONT	Changes the font and the font size of an element with text.
MOVE TO FRONT	Places the selected element into the foreground.
MOVE TO BACK	Places the selected element into the background.
MOVE UP	Moves the selected element up one step.
MOVE DOWN	Moves the selected element down one step.
FLIP HORIZONTAL	Flips the selected element horizontally (only for components; see also "Arranging Components" on page 43).
ROTATE	Rotates the selected element 90° around its rotation point (only for components; see also "Arranging Components" on page 43).
ALIGN	Aligns a group of selected elements with the **last** selected element (LEFT/RIGHT/TOP/BOTTOM)
SIZE	Defines SAME WIDTH, SAME HIGH, or BOTH for several selected elements. The size matches the last selected element.

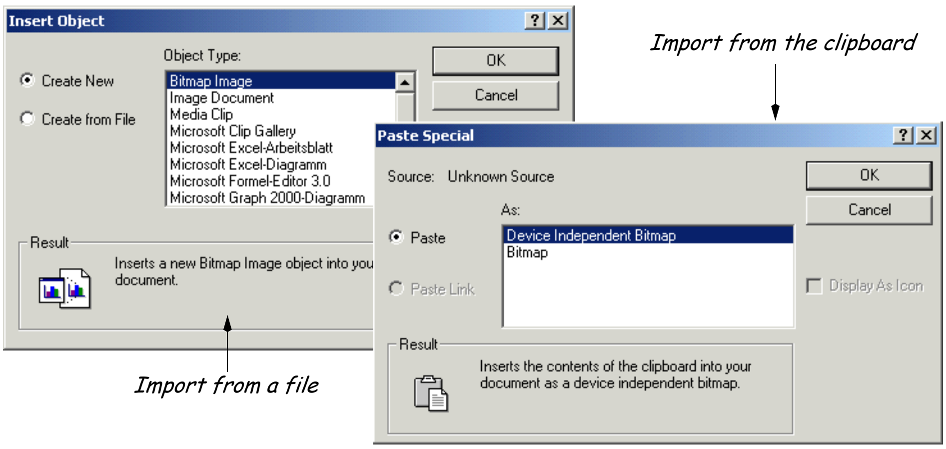
11.3 Data Exchange with Other Applications

SIMPLORER offers different features to import and export data. With the EDIT>INSERT NEW OBJECT command figures and text from other applications can be embedded into the Schematic sheet. Otherwise with the EDIT>COPY command you can copy Schematic data into the clipboard and make them available for other applications.

Importing Data

Beside internal drawing objects and elements, you can insert external data and graphics into the Schematic sheet. Schematic offers two functions to import data:

- the command EDIT>NEW OBJECT inserts an existing file or opens an application on your computer to create a new object
- the command EDIT>PASTE SPECIAL inserts data in specified formats from the clipboard.



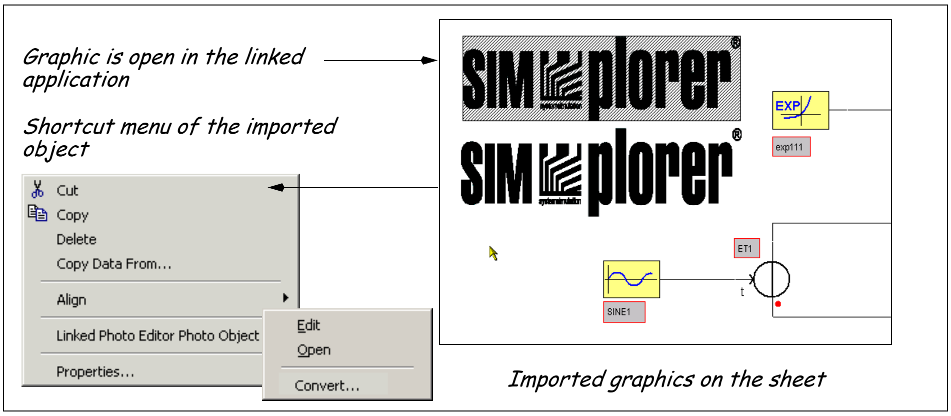
Displaying Links

The command EDIT>LINKS shows a list of all existing links to other applications in the sheet. You can update or delete the data from the sheet or open the linked application.

The EDIT>LINKS command displays only those files which are inserted in the Schematic with EDIT>INSERT NEW OBJECT and with the **«Link»** option checked.

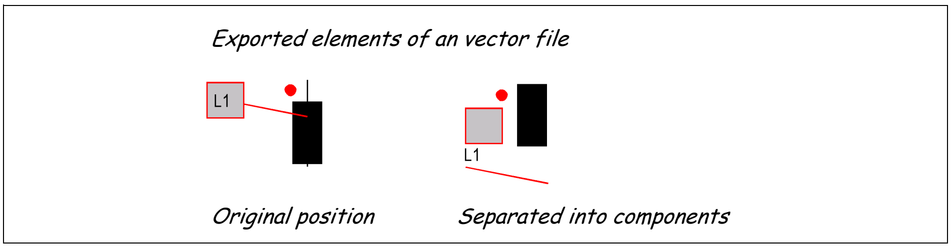
Editing Properties

You can start the application for inserted objects, edit the graphic and return to the Schematic. To change the size of an object, drag on the object handels with the mouse. If you press, in addition, the SHIFT key, you can adjusting the size proportionally.



Exporting Data

The command EDIT>COPY copies selected elements into the clipboard as a vector file (.wmf). Use EDIT>PASTE or EDIT>PASTE SPECIAL to insert the graphic into other Windows applications. If the target application supports vector graphics, you can separate the image into its components and modify the individual parts.



With the FILE>IMPORT and FILE>EXPORT commands, you can import or export data to and from other model sheets. See also 11.1 “File Management in Schematic” on page 339.

11.4 Arranging Screen Layout and Using Help

You can adapt the windows arrangement to their individual preferences. The following parts can be changed:

- The presentation of one model sheet (zoom in, zoom out, viewing layers)
- Windows arrangement of the model sheets (windows side by side, part windows, identical windows)

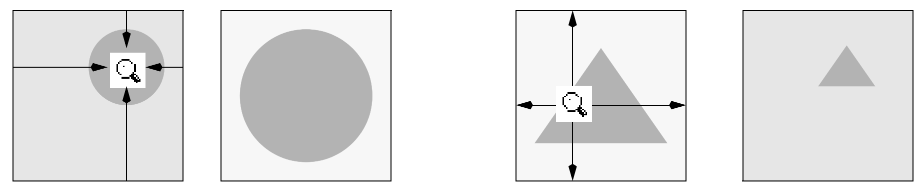
Display and arrangement of Schematic Tools (toolbars, model tree, information window).

Zoom Functions

You can change the visible area of the model sheet with the VIEW commands or the corresponding functions in the ZOOM toolbar. Especially for extensive sheets, you can access separate areas more quickly with VIEW commands.

<u>Zoom Function (Point zoom)</u>

The Point zoom function defines the zoom area with a mouse click. Click the symbol or select VIEW>POINT ZOOM to start the zoom mode. The mouse pointer becomes a "magnifying glass." Every click with the left mouse button in the sheet zooms in (enlarges) the representation of elements; every click with the right mouse button zooms out (reduces). The ESC key closes the zoom mode and the normal select mode is active again.



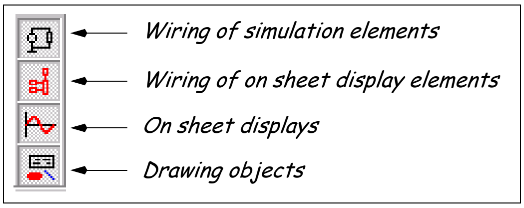
Click with the left mouse button, the area is zoomed in

Click with the right mouse button, the area is zoomed out

View Menu

Command	Toolbar Symbols and Description
ZOOM IN	Enlarges the active window by one step.
ZOOM OUT	Reduces the active window by one step.
ZOOM REGION	Enlarges the area selected with the mouse.
MARKED OBJECTS	Enlarges the selected elements to maximum size.
ALL OBJECTS	Enlarges the display of the entire model sheet to maximum size.

Command	Toolbar Symbols and Description
WHOLE PAGE	Displays the whole page of the model sheet.
POINT ZOOM	Zooms in (left mouse click) or zooms in (right mouse click); the selected point is moved into the window's center.
REFRESH	Updates the screen display.
VIEW LAYERS	Shows or hides elements with common properties. These commands are equivalent to some symbols in the Draw toolbar (as shown below).



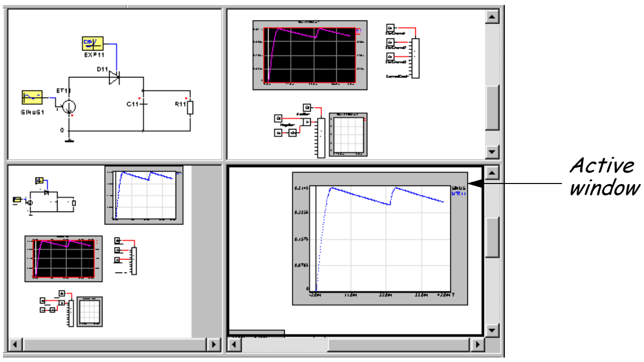
Command	Toolbar Symbols and Description
MODEL TREE	✓ Shows or hides the model tree.
OBJECT BROWSER	✓ Shows or hides the Object browser.
OUTPUT/MESSAGES	✓ Shows or hides the information window.
REPORT BROWSER	✓ Shows or hides the Report browser.
TOOLBARS	✓ Shows or hides the Schematic toolbars: SIMULATE, MODELS, ZOOM, DRAW, CONNECT, ARRANGE, MAIN, COMMAND BAR.
STATUS BAR	✓ Shows or hides the status bar with help messages and key status display.

Split Windows

Schematic can handle multiple sheets in multiple windows simultaneously. Window style and screen arrangement is managed by the WINDOW menu commands.

Windows Menu

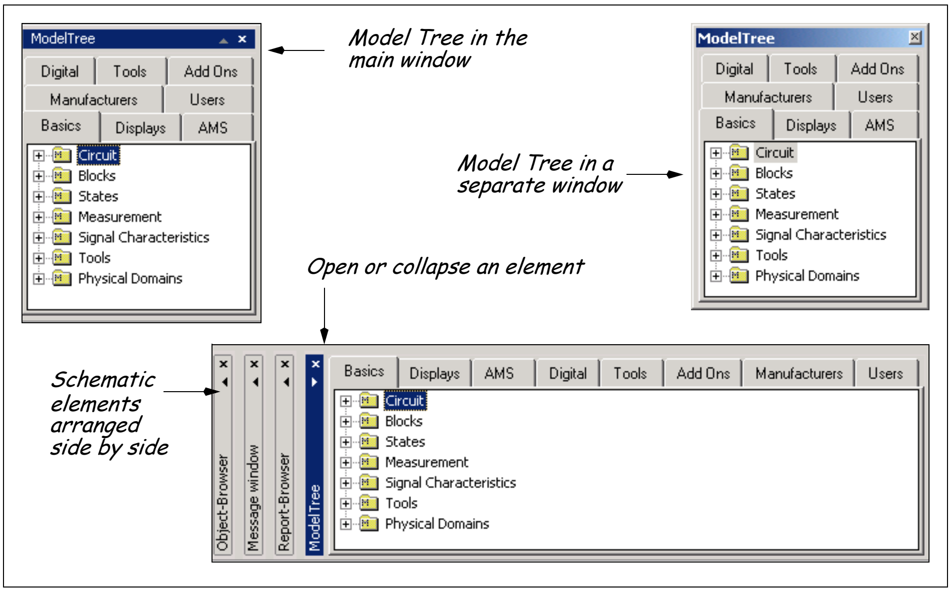
Command	Description
NEW WINDOW	Creates a copy of the active window. All wave forms in the original and duplicate window are identical. You can recognize the duplicate window by its name in the title bar [name.SSH:2].
SPLIT	Splits **one** sheet into 4 parts, which represent several details of the wiring. The partition can be removed by selecting and dragging the window partition cross to the border of the sheet window.



Command	Description
OVERLAPPING	Rearranges all open windows in a cascade. All title bars are visible.
TILE VERTICALLY	Rearranges all open windows side by side.
TILE HORIZONTALLY	Rearranged all open windows one on top of the other.
ARRANGE ICONS	Moves all minimized windows to the lower part of Edit window and arranges them.
«LIST OF OPEN WINDOWS»	Lists all open Windows and their corresponding file names.

Arranging Schematic Elements

The Schematic main window consists of several elements you can hide, display, and move to different places: Toolbars, Status Bar, Model Tree, Message window, Report Browser, and Object Browser. Each of these is managed in a separate pane. You can arrange the elements by dragging with the mouse at the double line on the left hand side. You can arrange a selected element within the main frame or drag it to a separate location.



Help and Version Information

Online help is available for all SIMPLORER programs, from the HELP menu or by clicking the symbol on the toolbar.

Help Menu ▾

Command	Toolbar Symbol and Description	
HELP CONTENTS		Go to online help.
TIP OF THE DAY		Opens help dialog for simple Schematic functions.
ABOUT SCHEMATIC		Product and version information.

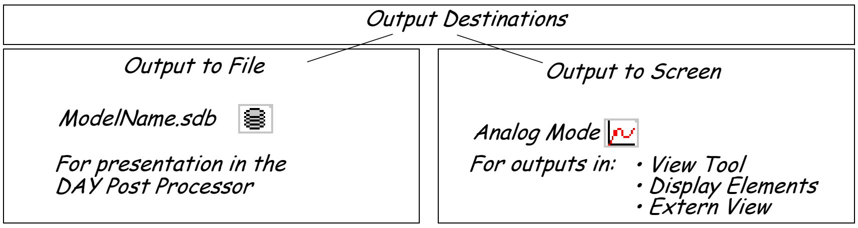
12 Simulation Results

This chapter contains information on:

- Representing simulation results
- Defining simulation outputs
- Using Display Elements
- Using the View Tool

12.1 Representing Simulation Results

A simulation results in several types of data, which can be saved in an output file or displayed online using output definitions.



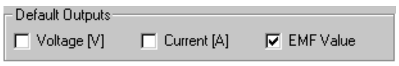
Display Elements and “Extern View” representation can be used only if the simulation model was created in the Schematic. See also 12.2 “Display Elements” on page 369.

Output Formats

SIMPLORER 6.0 programs automatically create .sdb files to store simulation results and data, but the .mdx, .mda, and .mdk formats from former versions can still be used. The .mdx format or other ASCII file formats are used to describe characteristics. See also “The .mdx File Format” on page 467.

Defines an analog online output for the View Tool.

Defines an offline output (.sdb file) for DAY Post Processor or external programs.



Defines an offline output (.sdb file) for DAY Post Processor or external programs.

Output Tools

Display Elements

Display Elements are a group of elements to display results online during the simulation directly on the sheet in the Schematic. Display Elements are included in the «**Display**» tab and are placed on the sheet like any other component. There are different types of Display Elements for displays in graphical, numeric, and table form as well as for signal processing as mathematical functions, data management, or FFT analysis.

Except from the Bode Plot and Nyquist Diagram, every Display Element can open its own External View display, which has, in addition, a cursor function for data analysis.

View Tool

The View Tool is a program to display results online during the simulation outside the Schematic. The View Tool is started whenever definitions for graphical online outputs are included in the simulation model and the Display Elements display are deactivated. If an AC or a DC simulation was carried out, real and imaginary part as well as magnitude and phase of a data channel are displayed in the View Tool.

DAY Post Processor

The DAY Post Processor is a program to display results offline after the simulation outside the Schematic. The DAY Post Processor uses .sdb result files created when definitions for file outputs are included in the simulation model.

Defining Outputs of Components

You can define component outputs in three ways:

- In the component dialog
- In Display Elements
- In the Model Sheet

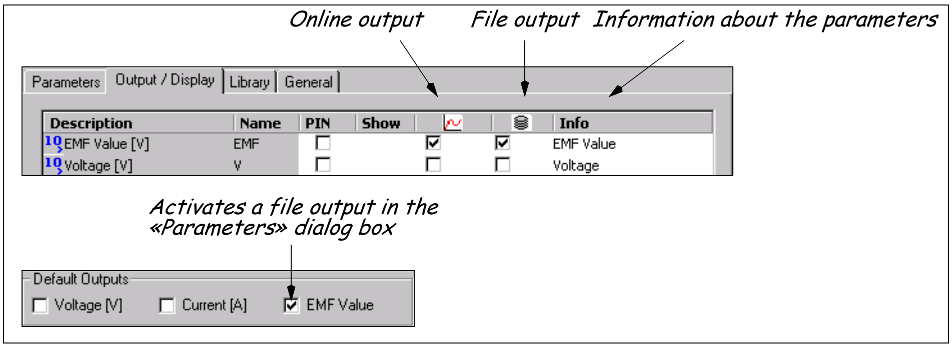
The representation, which can be either analog or digital, can be changed in the data channel properties in the output tools, which means you can also assign the proper format after running a simulation.

Outputs in Component Dialogs

Each component has an property dialog with an «**Output/Display**» tab, where you can define outputs and output formats. This tab lists all component parameters and outputs. To activate an output, select the check box for the corresponding entry. If you select «**Default Outputs**» in the «**Parameters**» tab, the selected simulation quantity is written to the .sdb file.

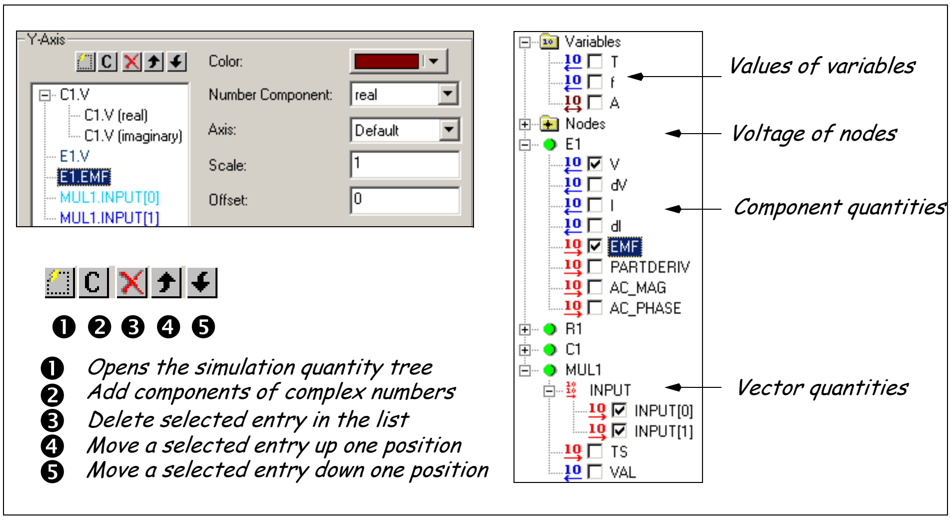
12

Double-click a component symbol and click the «**Output/Display**» tab. Select an online and/or file output of a simulation quantity.



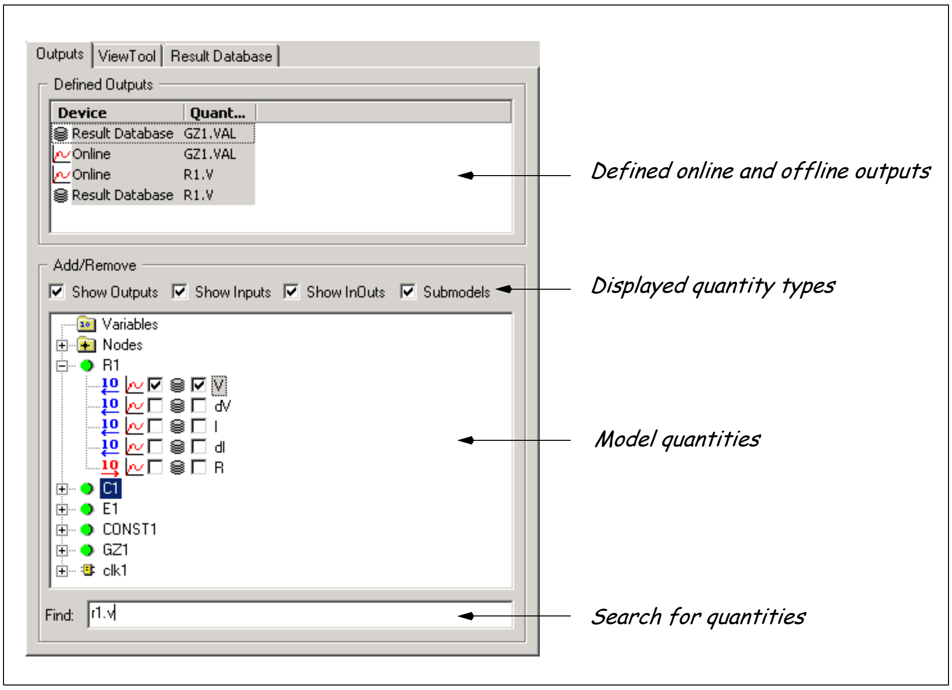
Outputs in Display Elements

Display Elements provide all simulation quantities of a sheet for online output. Place a Display Element on the sheet. Create a new output entry and select a parameter in the simulation quantity tree. For each defined output quantity you can set color, displayed number component (real, imaginary, magnitude, or phase), Y axis, scale, and offset. In Display Elements, you can define only online outputs. Number components are only available with an AC simulation. For transient and DC simulations only real components provides values.



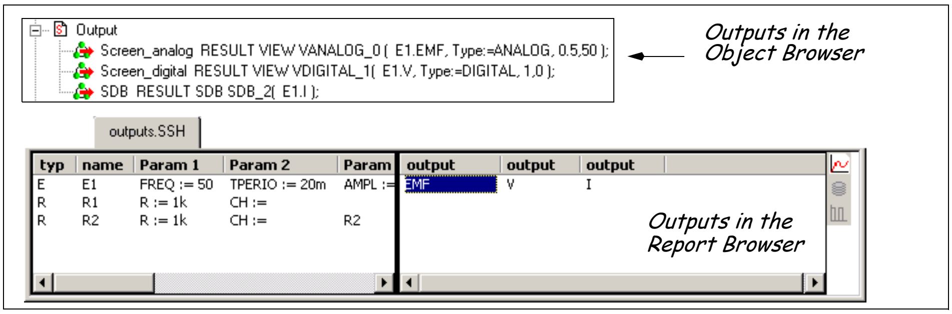
Outputs in the Model Sheet

When you choose SIMULATION>OUTPUTS, a tree appears showing all available output quantities. From this tree, you can access all models and submodels used in the sheet. To define an online output or file output, select a parameter from the simulation quantity tree.



Modifying Outputs of Components

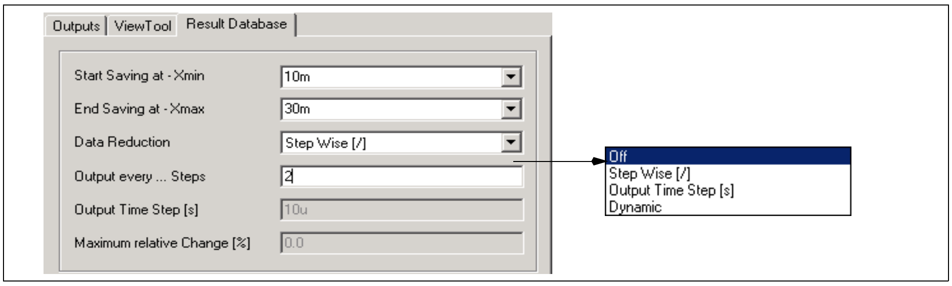
The Object Browser and Report Browser also display all outputs defined in a model sheet, and, in both browsers, you can delete defined outputs. While the object browser displays a separate folder for all outputs, the report browser displays the outputs next to components in the list.



Using Data Reduction for Simulation Results

SIMPLORER offers a powerful feature to reduce the size of output files. In the SIMULATION>OUTPUTS **«Result Database»** dialog, you can define options for saving simulation outputs in files. The following options are applicable to files:

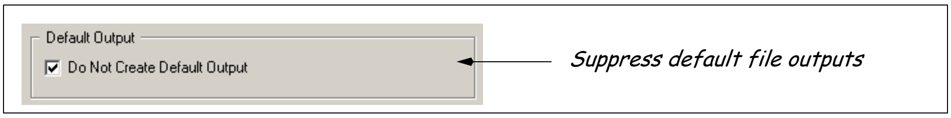
- Set the interval (start and end time) in which results are stored (start and end frequency for AC simulation)
- Reduce frequency of stored simulation steps: step wise, time dependent (frequency dependent for AC simulation), or dynamic (dependent on the value change related to the last value in percent, 0% means constant values are stored only one time)



Switching Off File Outputs

All file outputs, defined in a model sheet, are stored in one .sdb result file. By default, numerous component quantities have file outputs. Default outputs are defined in the model library for each component.Therfore, if you not clear these default output definitions after dropping, a large result file is created of each simulation run.

To suppress file outputs, check the option **«Do not Create Default Output»** in SHEET>PROPERTIES **«System»**. Additional file outputs, defined after dropping a component, are not affected from this option. To save online outputs in .sdb files, choose FILE>SAVE after a simulation run.



View Tool Coordinates

The settings in the SIMULATION>OUTPUT **«VIEW TOOL» dialog** define the representation of the View Tool coordinates. Please note: If you save the View Tool representation of a certain simulation model, these settings have a higher priority and are used for the display. See also 12.3.1 “Data Management” on page 386.

12.2 Display Elements

SIMPLORER’s Display Elements feature displays simulation results directly on the sheet and offers more options for presentation than the View Tool. The presentation capabilities range from graphical curves, numerical and table displays to mathematical functions. Display Elements can be embedded in the model description only with the Schematic.

Display Elements appear in the **«Display»** tab and are placed on the sheet like any other component. In the model tree, Display Elements are identified by a red triangle beside the name. The simulation outputs, layout, and presentation of Display Elements are defined in the property dialog. Do one of the following to open the property dialog:

- double-click the Display Element,
- click with the right mouse button the Display Element and choose **«Properties»** or,
- choose the EDIT>PROPERTIES command for the selected Display Element.

When the simulation is finished, simulation quantities can be added or removed without starting the simulation again. Size and location of the display elements can be adapted individually. The VIEW>VIEW LAYERS>ACTIVE WIRING command switches Display Elements On or Off.

Once you have placed an Display Element on the model sheet, the element is used for graphical online output. To see outputs in the View Tool, select the option **«Use View Tool for Simulation Output»** in the dialog SHEET>PROPERTIES **«Systems»**.

Most of the property dialogs have a **«Use as Default»** option. To save current settings of a page as default, check the option and click <Apply>. To restore changed settings, click <Default> and all settings on a page are set to default values.

For all Display Elements there are the **«General Settings»** tab. You can select how often the graphic is updated during a simulation. Select **«Each Simulation Time Step»**, **«50% of Simulation End Time»**, or **«Simulation End Time»**. Reducing the update frequency is useful for large simulation models to save simulation time. For all Display Elements there are the **«Library»** tab. Here the path, library, and model identifier are displayed.

12.2.1 View Elements

View Elements displays simulation results graphically, digitally, or in table format. There are two ways to assign simulation quantities for output. For Quick View Elements, the assignment is made through a definition within the property dialog, for Connect View Elements, the assignment is made through simulator channel elements.

Outputs in Quick View Elements

Display Elements provide all simulation quantities of a sheet for online output, including simulation time “t” and frequency “f”.

To assign a simulation quantity for the Y axis do the following:

1 Place a display element on the sheet.

2 Double-click the symbol to open the property dialog.

3 Select either «Time Channel» or «Frequency Channel» as X axis value.

a. To assign another quantity as X axis value, select «Others» and click the symbol.

b. Select one of the parameter in the simulation quantity tree and click <OK>.

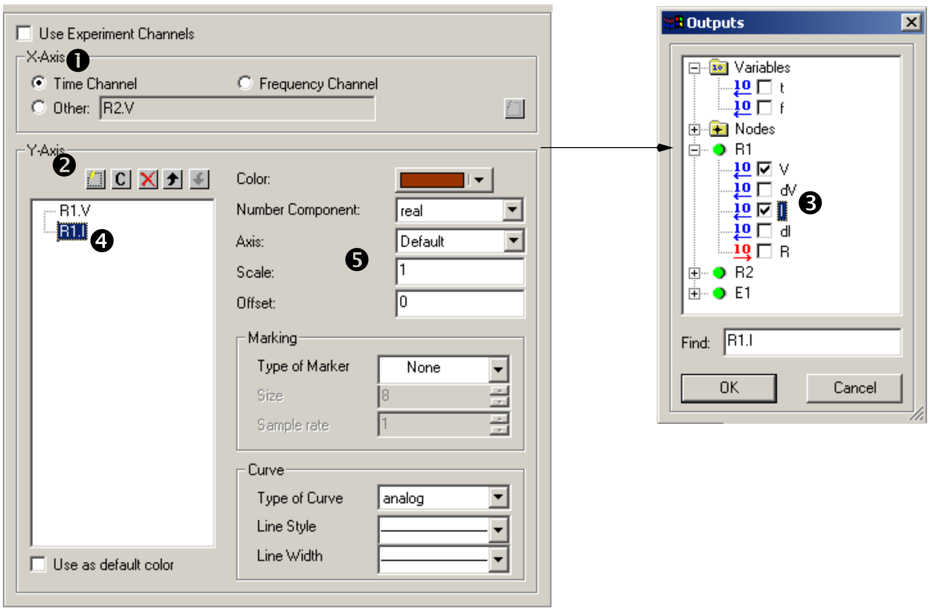
4 Click the symbol in the Y axis area.

5 Select all parameters you want in the simulation quantity tree and click <OK>.

a. To add real, imaginary part, magnitude, and/or phase for output, select the quantity in the output tree, click the C symbol and select one of the entries. If you want to change the current number component, select an entry in the «Number Component» list.

Number components are only available with an AC simulation. For transient and DC simulations only real components provides values.

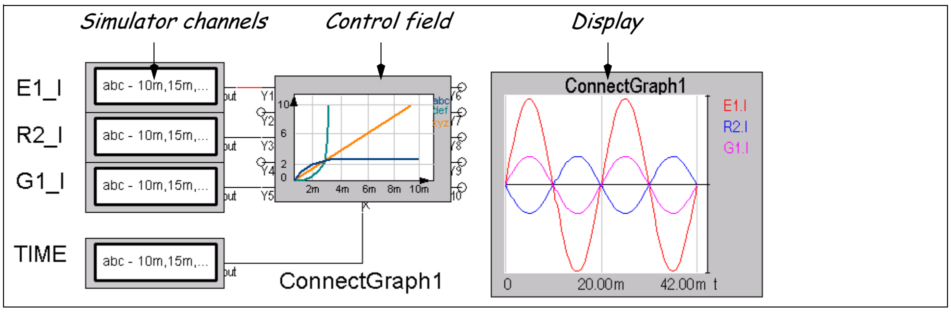
b. To define color, Y axis, scale, and offset, select the quantity in the output tree and define the settings in the dialog.



❶ *Select the simulation quantity of the X axis*
❷ *Click the "New entry" symbol to open the simulation quantity tree*
❸ *Select the simulation quantities for output and click <OK>*
❹ *Select the entry in the tree of output quantities*
❺ *Make the settings you want in the dialog*

Outputs in Connect View Elements

The starting point for wiring Connect View Elements is the element's simulation channel, to which a simulation quantity from the model is assigned. The outputs of the simulation channels are connected with views, tables, or operators. The outputs of operators are connected in turn with a display of a connect element. The X axis usually represents the simulation time, but it may represent any other element. Only simulation quantities connected with the Connect View Element are available for output.



Connect Elements have both the graphic display and a control field, which receives connections from the simulator channels. The Connect element form is especially meaningful for output values that are displayed after mathematical operations.

The wiring assigned with Display Elements does not influence the actual simulation model. The simulator channels are used only to determine the graphical presentation.

If the control field is deleted, the display is also deleted.

Connect View Elements have an **«Output/Display»** and **«Parameters»** dialog. The settings in these dialogs have no influence on the function itself.

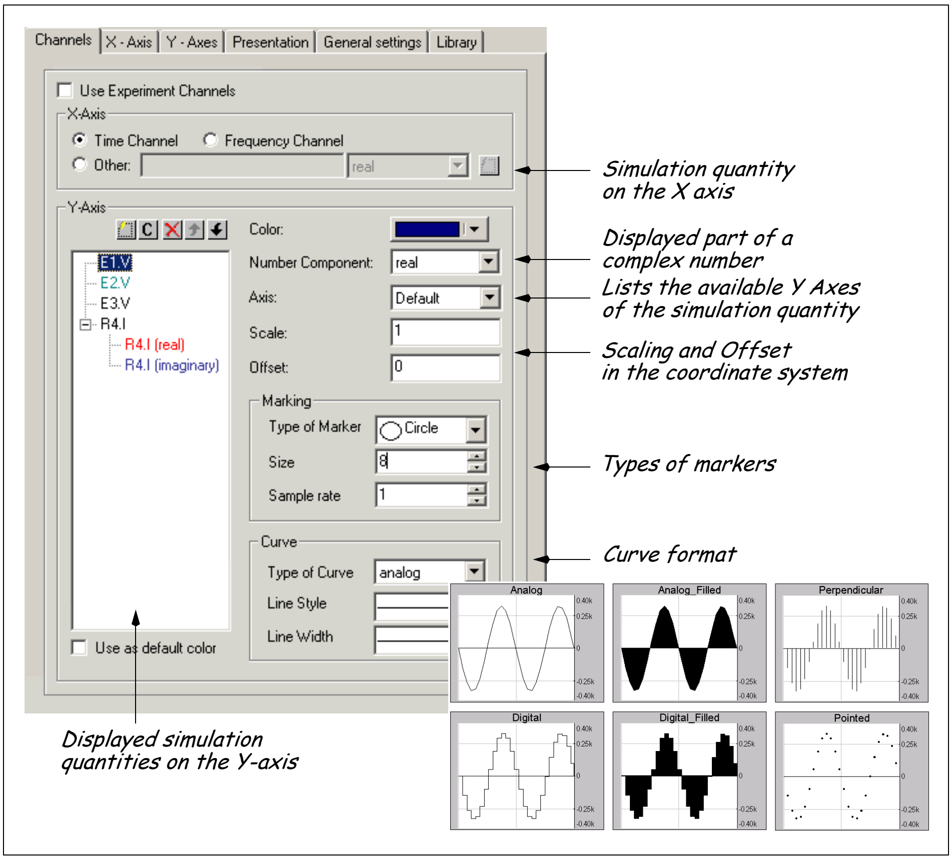
2D View Element

The Display Elements 2D View and 2D View (con) show the graphical curves of the assigned simulation quantities.

Channels Tab

The assignment of simulation quantities for the X and Y axes is defined in the **«Channels»** tab. All defined simulation quantities, the system time t, and the frequency f for both X and Y axes are available.

You can define color, marker, representation type, and line of the curve. If more than the default Y axis is available, you can select a certain Y axis for the simulation quantity from the list in the Axis column. Additional Y axis are defined in the Y Axis dialog. Also, a scale and shift factor in Y-direction can be entered to present curves of different dimensions in the same coordinate system. See also "Outputs in Quick View Elements" on page 370 and "Outputs in Connect View Elements" on page 371.



X Axis Tab

Sliding Window

The Sliding Windows defines the maximum display of the X range. The window will be moved during the simulation between the start and simulation end time.

Presentation

The settings define position, color, and the font of the inscription of the X axis.

Scale

You can set Linear or Logarithmic representation for the X axis.

The adaptation of presentation areas for the X and Y axes can be performed manually or automatically. If automatic adaptation is selected, the simulation quantities are displayed at the maximum size. The dimension defines the maximum extended channel. If automatic values are not selected, the values entered for the axes scales are valid.

Grid Lines

You can define if grid lines are displayed in the diagram and the minimum number.

Data Format

You can select three different formats (12.3k/12.3E+3/1.23E+4) from the list and the number of decimals (**«Fixed Mantisse»**).

Y Axis Tab

Add/Delete Axis

The 2D Display Element supports multi axes graphic. To assign curves to a special coordinate system with its own scaling you have to define a separate Y axis. Select **«New Y Axis»** from the list and type a name and when necessary an additional information in the input fields below. Click <Add Axis> to create a new entry for an Y axis. All settings displayed in the dialog are applied to the Y axis selected in the drop-down list.

Presentation

The settings define position, color, and the font of the inscription of the Y axis. The value of **«Fixed Width»** assigns a fixed value for the display in the diagram.

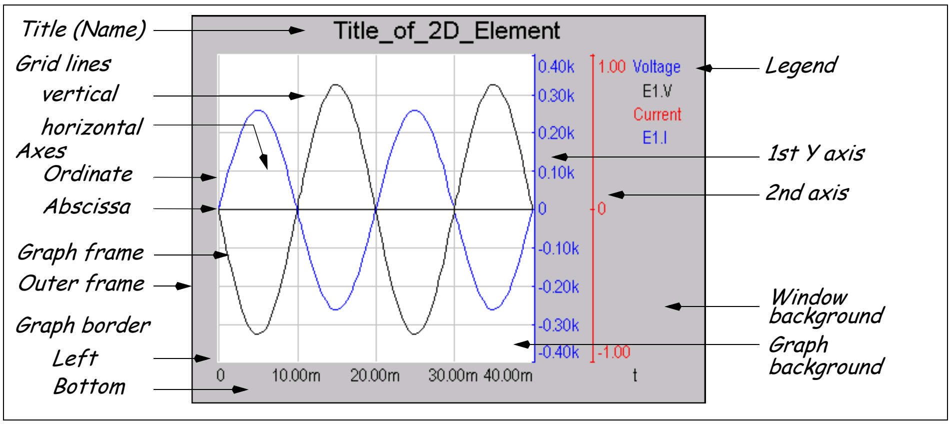
Scale/Grid Lines/Data Format see X axis.

Presentation Tab

The Presentation dialog defines all properties of the diagram's appearance. If a check box is selected, the corresponding option is active. The name, defined in the text box, is used as title in the representation on the sheet.

Graph Border

The graph border settings define the space between outer and inner frame of the Display in percent.



Color

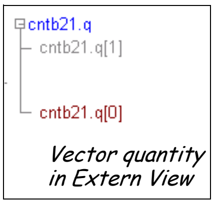
The colors in the dialog define the presentation on the model sheet. Click the corresponding color box to open the color menu to change the color.

Color Scheme

Schematic also offers predefined color schemes or the option to save user-defined color schemes. To save your own color scheme set the colors in the dialog, type a new name for the scheme in the input field and click <Save>. You can load an existing scheme when you select a name from the list and click <Load>.

The selected colors are also used for printing. Black backgrounds are not recommended.

2D Digital Graph



The 2D Digital Graph and 2D Digital Graph (con) are especially designed to display quantities of VHDL-AMS models. In contrast to the 2D View, each quantity has its own coordinate system. The 2D Digital Graph is able to show vector quantities, special display formats, and data types. To show and hide members of a vector, you have to open the «Extern View» and click the + sign in the diagram legend.The 2D Digital Graph has a special cursor display in the Extern View. At a glance, you can see all quanties referring to the simulation time.

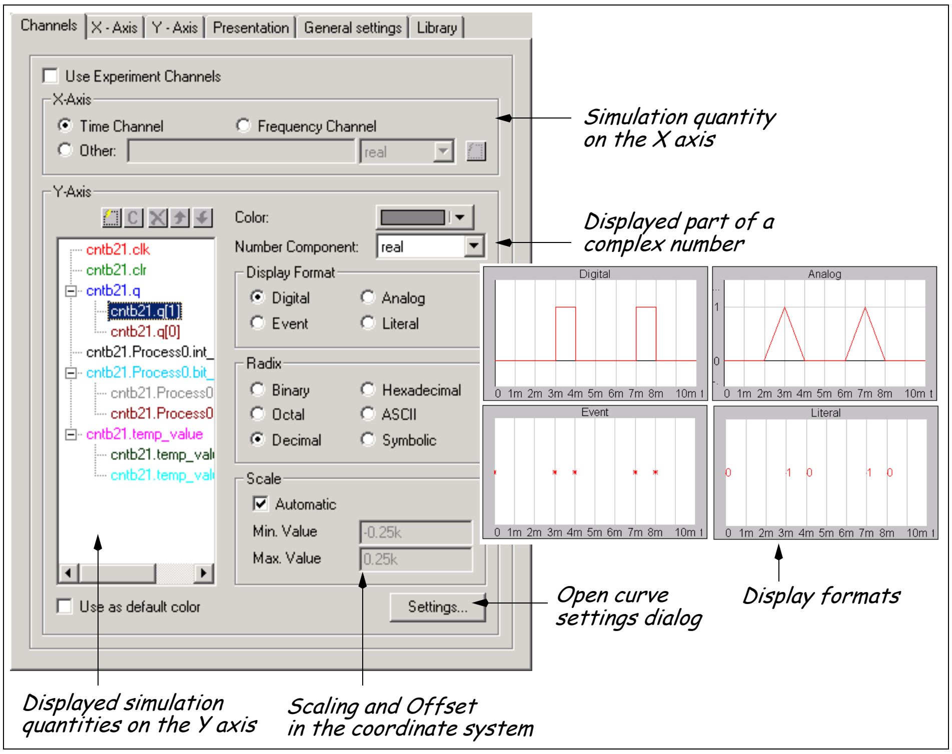
Channels Tab

The assignment of simulation quantities for the X and Y axes is defined in the **«Channels»** tab. All defined simulation quantities, the system time t, and the frequency f for both X and Y axes are available.

Depending on the data type and the display format you can define different data formats for representation (Radix). The formats are available only for Literal display formats. The Symbolic representation is used for ENUM types. The decimal number 100 would be represented as 64 in hexadecimal format, as 144 in octal format, as 0110 0100 in binary format, and as d in ASCII format. Zoom in the graphic when literals are not displayed in the view.

To define color, marker, representation type, and line of the curve, click <Settings>. Also, a scale and shift factor in Y direction can be entered for a quantity. See also “Outputs in Quick View Elements” on page 370 and “Outputs in Connect View Elements” on page 371.

12



X Axis Tab

Sliding Window
The Sliding Windows defines the maximum display of the X range. The window will be moved during the simulation between the start and simulation end time.

Presentation
The settings define position, color, and the font of the inscription of the X axis.

Scale
The adaptation of presentation areas for the X- and Y-axes can be performed manually or automatically. If automatic adaptation is selected, the simulation quantities are displayed at the maximum size. The dimension defines the maximum extended channel. If automatic values are not selected, the values entered for the axes scales are valid.

Grid Lines
You can define if grid lines are displayed in the diagram and the minimum number.

Data Format
You can select three different formats (12.3k/12.3E+3/1.23E+4) from the list and the number of decimals.

Y Axis Tab

Presentation/Grid Lines/Data Format see X axis.

Presentation Tab

The Presentation dialog defines all properties of the diagram's appearance. If a check box is selected, the corresponding option is active. The name, defined in the text box, is used as title in the representation on the sheet.

Graph Border
The graph border settings define the space between outer and inner frame of the Display as well as diagrams in percent. The value in the «Member» box set the distance between members of a vector.

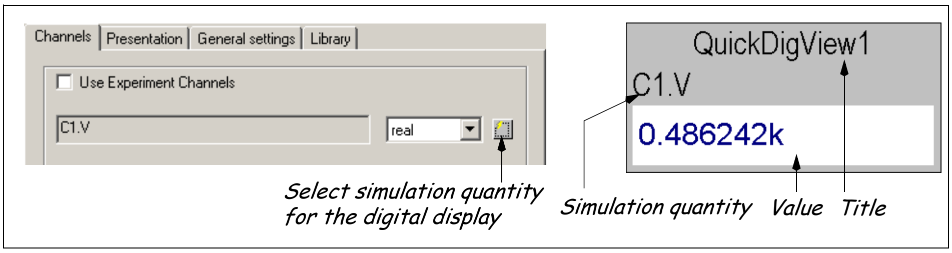
Diagram Height
The values define the height of the corresponding diagram type (analog, digital, event, literal) in percent.

Color
The colors in the dialog define the presentation on the model sheet. Click the corresponding color box to open the color menu to change the color.

Color Scheme
Schematic also offers predefined color schemes or the option to save user-defined color schemes. To save your own color scheme set the colors in the dialog, type a new name for the scheme in the input field and click <Save>. You can load an existing scheme when you select a name from the list and click <Load>.

Numerical View Element

The Display Elements Numerical View and Numerical View (con) show the active value of the assigned simulation quantity.



Channels Tab

For Connect elements, the assignment is made through a simulator channel element. Only one output channel can be selected for the display. See also “Outputs in Quick View Elements” on page 370 and “Outputs in Connect View Elements” on page 371.

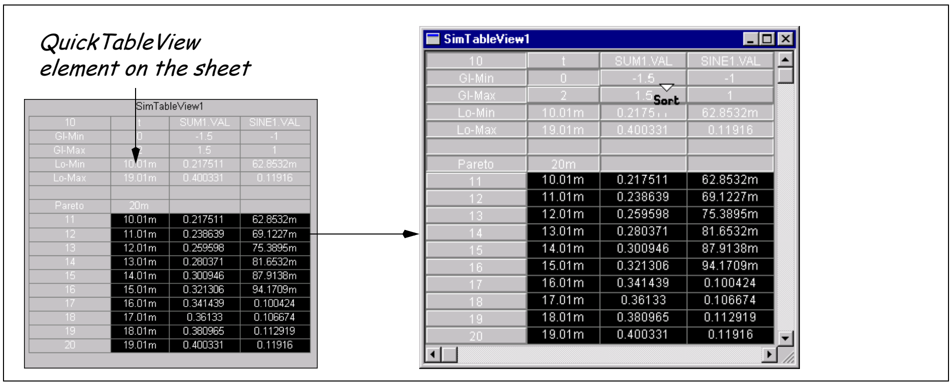
Presentation Tab

The Presentation dialog defines all properties of the displays’s appearance. If a check box is selected, the corresponding option is active. The name, defined in the text box, is used as title in the representation on the sheet.

Table View Element

The library of Display Elements has an element to represent simulation data in table form (Table View). In the **«Extern View»** mode, you can apply all known functions (sort data sets, apply filter, create Pareto set) of the DAY Optim Post Processor.

You can copy selected data sets with CTRL+C into the clipboard and after that insert the data as formatted Text (Tab and CR) into a text editor. In the **«Extern View»** mode you can scroll the data sets in the window and use DAY Optim commands. See also “DAY Optim Post Processor” on page 472.



Channels Tab

The assignment of simulation quantities for the X and Y axes is defined in the **«Channels»** tab. All defined simulation quantities, the system time t, and the frequency f for both X and Y axes are available.

See also “Outputs in Quick View Elements” on page 370 and “Outputs in Connect View Elements” on page 371.

Presentation Tab

The Presentation dialog defines all properties of the diagram’s appearance. If a check box is selected, the corresponding option is active. The name, defined in the text box, is used as title in the representation on the sheet.

Color

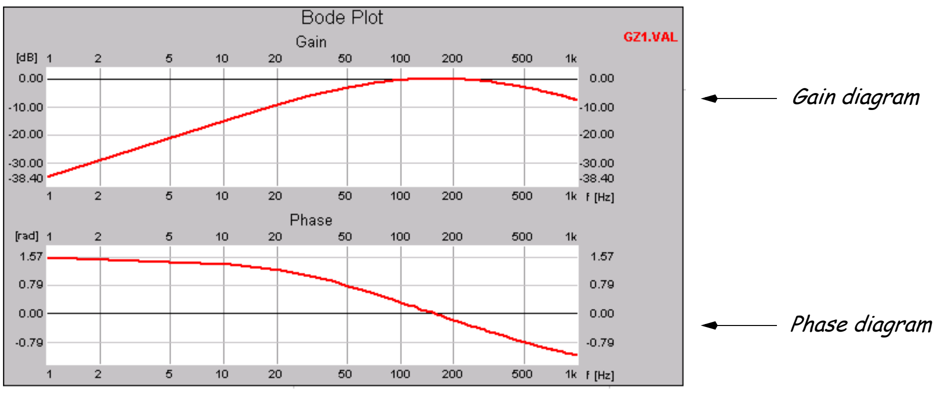
The colors in the dialog define the presentation on the model sheet. Click the corresponding color box to open the color menu to change the color.

Color Scheme

Schematic also offers predefined color schemes or the option to save user-defined color schemes. To save your own color scheme set the colors in the dialog, type a new name for the scheme in the input field and click <Save>. You can load an existing scheme when you select a name from the list and click <Load>.

Bode Plot

The Bode Plot and Bode Plot (con) are especially designed to display quantities of AC simulations. In contrast to Display Elements, the Bode Plot has not Extern View function. For a defined quantity gain and phase are represented; each in a single diagram in the Bode Plot.



Channels Tab

The assignment of simulation quantities for the X and Y axes is defined in the **«Channels»** tab. All defined simulation quantities, the system time t, and the frequency f for both X and Y axes are available. The frequency is set by default for the X axis.

You must define at least two quantities (signals) for the Y axis; the investigated output signal and the base channel. Define the output channel, select the quantity in the tree and click the new entry symbol to define the base channel.

To define color, marker, representation type, and line of the curve, click the <Settings> button. Also, a scale and shift factor for gain and phase can be entered for a quantity.

See also "Outputs in Quick View Elements" on page 370 and "Outputs in Connect View Elements" on page 371.

Presentation Tab

Display

The Presentation dialog defines all properties of the diagram's appearance. If a check box is selected, the corresponding option is active. The names, defined in the text box, are used as titles in the representation on the sheet.

Distances

The graph border settings define the space between outer and inner frame of the Display as well as the gain and phase diagram in percent. The value in the «Legend» box set the distance for the legend.

Legend

Defines if the legend is displayed in the diagram.

Frequency Axis Tab

Vertical Grid Lines

You can define if grid lines are displayed, color, representation type, line width, and number.

Frequency Axis Description

You can define if the description is displayed, color, font, and the description text.

Frequency Axis Inscription

You can define if the inscription is displayed, position (bottom or top), color, font, format, mantisse, and unit (frequency or angular frequency). You can select three different formats (12.3k/12.3E+3/1.23E+4) from the list and the number of decimals (**«Fixed Mantisse»**).

Frequency Axis Scale

The adaptation of presentation areas for the axis can be performed manually or automatically. If automatic adaptation is selected, the simulation quantities are displayed at the maximum size. The dimension defines the maximum extended channel. If automatic values are not selected, the values entered for the axes scales are valid.

Gain Axis Tab

Horizontal Grid Lines

You can define if grid lines and the axis are displayed, color, representation type, line width, and number.

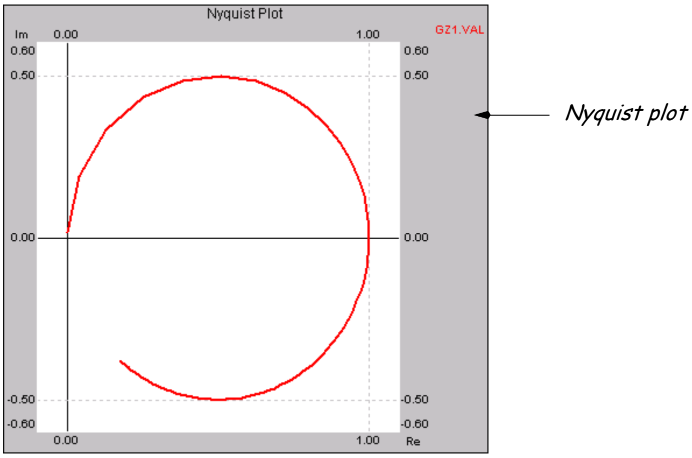
Gain Axis Description/Gain Axis Inscription/Gain Axis Scale see frequency axis.

Phase Axis Tab

Phase Grid Lines/Phase Axis Description/Phase Axis Inscription/Phase Axis Scale see gain axis.

Nyquist Plot

The Nyquist Plot and Nyquist Plot (con) are especially designed to display quantities of AC simulations. In contrast to Display Elements, the Nyquist Plot has not Extern View function.



Channel Tab

The assignment of simulation quantities for the X and Y axes is defined in the **«Channels»** tab. All defined simulation quantities, the system time t, and the frequency f for both X and Y axes are available. The frequency is set by default for the X axis.

You must define at least two quantities (signals) for the Y axis; the investigated output signal and the base channel. Define the output channel, select the quantity in the tree and click the new entry symbol to define the base channel.

To define color, marker, representation type, and line of the curve, click the <Settings> button. Also, a scale and shift factor for gain and phase can be entered for a quantity. See also "Outputs in Quick View Elements" on page 370 and "Outputs in Connect View Elements" on page 371.

Presentation Tab

See "Presentation Tab" on page 378 of Bode Plot.

Real Axis Tab

See "Frequency Axis Tab" on page 378 of Bode Plot.

Imaginary Axis Tab

See "Gain Axis Tab" on page 378 of Bode Plot.

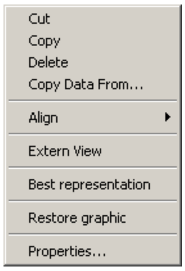
Probe Elements

Probe Elements are Display Elements which can assigned to a component or wire directly. Probes can display only one quantity. However, you can place any probes at each component you like. See also “Connect Toolbar and Menu” on page 48.

To place a probe, choose CONNECT>PROBE (the cursor changes into the probe symbol) and click a component or wire on the sheet. The default quantity, defined in the model library, is displayed, however, you can change it by selecting a new entry on the probe’s shortcut menu. If you place a probe at a wire, the corresponding across quantity is displayed.

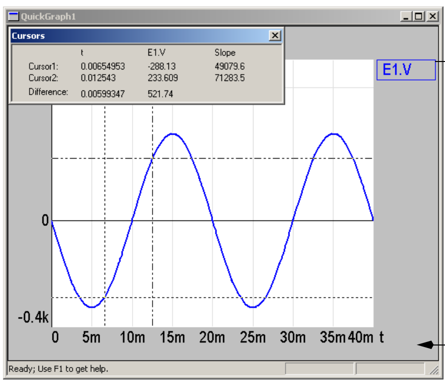
You can change the representation (digital or graphic) with the shortcut command **«Show as graphic»** or **«Show as digital view»**. In the digital view you can also change the number component (real, imaginary, phase, magnitude) with the shortcut menu commands.

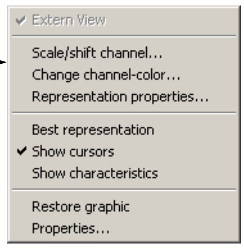
Display with Extern View



The **«Extern View»** command displays the outputs of an Display Element in a separate window outside the model sheet. This presentation mode is started with **«Extern View»** in the shortcut menu. In Extern View mode you can display both the absolute value of the coordinates and also the difference between two cursors on both the X and Y axes.

The cursors can be switched On with the shortcut menu (click in the graphic background with the right mouse button). Two cursors are displayed in the graphic window. The cursors can be moved by holding down the left mouse button (for cursor 1) or the right mouse button (for cursor 2). If you click another data channel name in the diagram or in the legend of the Y-axis, the cursor for this curve is activated. Close the window to finish the Extern View mode.





Shortcut menu of data channels with cursor function

Display in Extern View with activated cursors

Exporting Extern View Displays to other Applications

The key combination CTRL+C copies the active Extern View Window into the Windows clipboard as a metafile. Applications such as MS Word or Corel Draw can open the metafile (EDIT>PASTE SPECIAL) and edit certain parts of the graphic, such as curves, titles, or axis legends, as separate objects. This function is helpful if data presentations must be modified.

12.2.2 Postprocessing with Display Elements

The Display Postprocessing Elements provide online analysis functions. The DAY Post Processor, in contrast, can only use simulation results contained in files.

You cannot use results of the postprocessing elements in the simulation model. The display channels are used only to determine the graphical presentation.

Operations with Output Channels

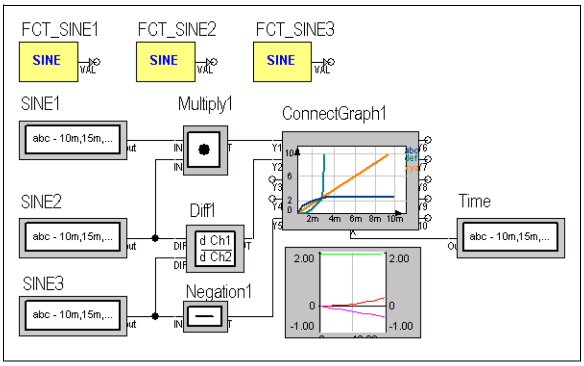
SIMPLORER offers several mathematical functions to perform signal post processing online on the sheet. Connect Views can display calculated values via a connection.

The outputs of the Simulation channels are connected with operators, and the output of the operator is connected in turn with a display element (through Connect View).

See also “Outputs in Connect View Elements” on page 371.

The following operations are possible:

- For one simulation channel
  Derivative
  Integration
  FFT
  Negation
  Reciprocal value
  Offset
  Scale
  Smooth
- For two simulation channels
  Multiplication
  Summation
  Channel Calculator



FFT Element

The FFT Element carries out an online FFT of an input signal connect to the pin and provides the result at the five output pins. In the sense of SIMPLORER Display Elements the FFT Element represents a connect element. The calculation is sine-based that means, a cosine signal would have a phase shift of $\pi/2$.

Inputs
I1: Input Signal, I2: External Time Channel

Outputs
Modulus and Phase
O1: Frequency, O2: Input Signal, O3: PHI [rad], O4: PHI [deg], O5: Amplitude [%]

Real and imaginary part
O1: Frequency, O2: Input Signal, O3: Imaginary Part

Base Channel
The choice **«Time Channel»** uses the simulation time. If you want to use a other channel, you have to select **«External»** and connect a simulator channel to input pin 2.

Time Window
The **«Automatic»** option carries out the calculation the complete simulation time. If the box is clear, you can define your own calculation window.

Filter Amplitude [%]
Amplitudes smaller than the defined proportional value of the maximum are not displayed.

Percent Evaluation
If the box is selected, the maximum amplitude value is set to 100%; otherwise the value entered in the field is used.

Result
Select **«Absolute Value and Phase»** or **«Real and Imaginary Part»** for the representation of the results.

Window
Select the Rectangle, Triangular, Von-Hann, Hamming, Blackmann, Lanzcos, or Weber window from the list. Select the rectangular window if you want to investigate a periodical signal.

Channel Calculator

The Channel Calculator carries out mathematical operations with quantities connect to the pins and provides the result at one of the five output pins. In the sense of SIMPLORER Display Elements the Channel Calculator represents a connect element.

Name
Defines the channel name used in Connect View Elements.

Expression
Click the expression field and open the edit dialog (…) to define mathematical expressions. You can use only quantities connected to the input channels. Double-click a input name in the calculator dialog list to insert the name in the input field.

12.2.3 Other Elements

In Addition to the View and postprocessing Display Elements there are the File Launcher and Data Channels. The File Launcher is used to start other applications when a defined condition becomes true. Data Channels provide a quantity of the simulation sheet to an Connect Display Element. Besides you can save simulation data of Display Elements in a .mdx file.

File Launcher

The File Launcher starts a specified file when a certain condition becomes or is true. You can select the frequency of starting with the Start Mode. The Connect Element has five inputs to connect channels with quantities. In the expression dialog you can select connected channels with a double-click.

Expression
Create, delete or move a entry with the icons placed on the upper right side. Click the expression field and open the edit dialog (…) to define a logical expression. You can use all quantities used in the simulation model. In a connect element you can use only quantities connected to input channels. Double-click the quantity/input name in the edit dialog list to insert the values in the input field.

File

Click the file field and open the file dialog (…) to define a file name. You can use all files existing on your PC. Guarantee that a proper application is linked with the file name.

Start Mode

Click the Start Mode field and select **«Always»**, **«Once»**, or **«At Crossing»** in the list. Always means at each simulation step when the logical expression is true.

The start of a file linked with a application takes a little time, so there is a delay from the start to the opened file. If you use **«Always»** be prepared for a lot of applications, which will be opened during the simulation.

Data Channels

Data from Simulation

The simulator channel element provides one simulation quantity of the model sheet for a Connect element. The simulator channel can be connected with any input from an Connect Element, for example with a postprocessing element. You can define the quantity in the **«Channel»** tab of the element dialog and connect the output to a input pin of a Connect Display Element. See also “Outputs in Quick View Elements” on page 370 and “Outputs in Connect View Elements” on page 371.

The channel element is required to provide simulation data for Connect Display Elements.

Data to File/Data to File Connect

The element saves simulation quantities, defined or connected to the element, in a .mdx file. You can save data as unformatted or formatted .mdx file or only the final values of the selected channel. The formatted .mdx files is legible from each text editor; the unformatted only from a SIMPLORER tool. See also “The Save Data Command” on page 386.

See also “Outputs in Quick View Elements” on page 370 and “Outputs in Connect View Elements” on page 371.

Data from File

The element provides simulation results stored in the following files:

- .sdb files
- .mdx files
- .xls files

Define a source file, select a output channel (01-10), click the new entry symbol and choose a quantity. You can select all quantities available in the file. In the **«Name»** box you can enter a description for the channel. See also “Outputs in Quick View Elements” on page 370 and “Outputs in Connect View Elements” on page 371.

12.2.4 Additional Functions

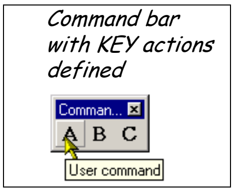
Display Elements have many different functions and forms. You can use them to display simulation results directly on the sheet and also to illustrate the simulation process with Animated symbols.

The Experiment tool provides results for Display Elements, if you select the Experiment Option.

Key Control with Display Elements

You can force events to occur during a simulation by using KEY action, i.e., pressing a predefined key.

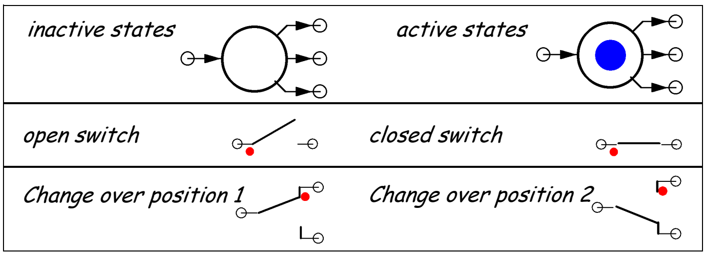
When the simulation is started, a toolbar is displayed with all KEY actions defined in a state graph. Click the corresponding symbol (letter) to set the mark in the state graph and start the actions. See also “Setting States with Key Actions” on page 244.



Animated Symbols

SIMPLORER’s Animated Symbols feature illustrates the operating sequence on the sheet depending on the value of a simulation quantity. The symbols change shape according to the active state of the element. Through these symbols the operating sequence of models can be understood more clearly and easily. When the simulation is complete, all symbols are reset to their initial state. The following elements have animated symbols:

- States
- Switches
- Transfer Switches



To display the states with animated symbols, select the **«Use Animated Symbols»** option FROM OR ON the SHEET PROPERTIES **«System»** menu.

If the simulation is interrupted, the “Animated Symbols Element” must be initialized. The command SIMULATION>RESTORE ANIMATED SYMBOLS resets the symbols to their initial state.

If you use simulation models from earlier versions of SIMPLORER, you must update the components to obtain the new functionality. The update can be started as the file opens or later with the command SHEET>SYNCHRONIZE. For a separate component, you can apply the **«Synchronize»** command from the shortcut menu of the element. See also “Updating Model Sheets from Former Versions” on page 342.)

Data Output from the Experiment Tool to Display Elements

Simulation data from the Experiment tool can be dragged into the model sheet to Display Elements by selecting the option in the Elements property menu: ☑ **«Use Experiment Channels»**.

12.3 View Tool

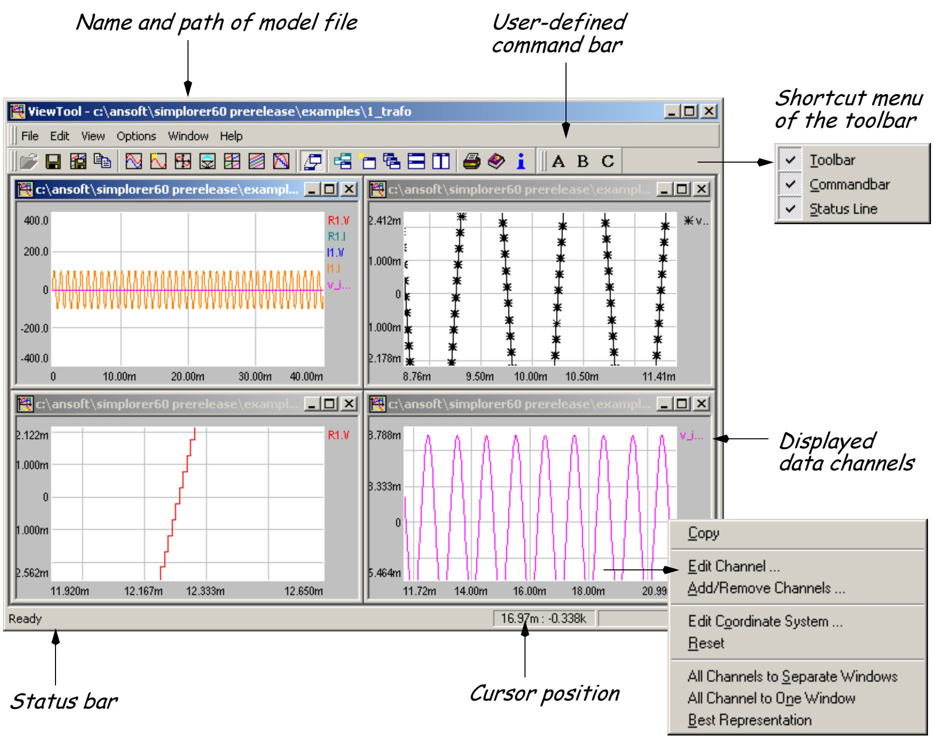
In addition to the Display Elements, the View Tool also presents simulation results in graphical form. The View Tool is started whenever definitions for graphical online outputs are included in the simulation model and the Display Elements display are deactivated. If an AC or a DC simulation was carried out, real and imaginary part as well as magnitude and phase of a data channel are displayed in the View Tool.

This section contains information on:

- Components of the View Tool main window
- Managing files and export file formats
- Editing data channels
- Settings for the coordinate system
- Settings for graphical representation and printing
- Windows arrangement and help

The View Tool Main Window

The waveforms specified for graphical output are displayed immediately during model calculation.



The View Tool provides multiple way to present simulation results:

- Using linear and logarithmic scales in coordinate systems
- Scaling and offsetting for each data channel
- Displaying data channel properties and characteristics
- Storing data in separate files
- Adding and removing of specified data channels
- Selectively zooming in on the coordinate system

If you are using the Display Elements in a sheet, the View Tool does not start until you select **«Use View Tool for Output»** in the SHEET>PROPERTIES **«System»** dialog.

12.3.1 Data Management

The Data Management menu commands store the waveforms (data channels) displayed in the immediate graphical output and also control the printer.

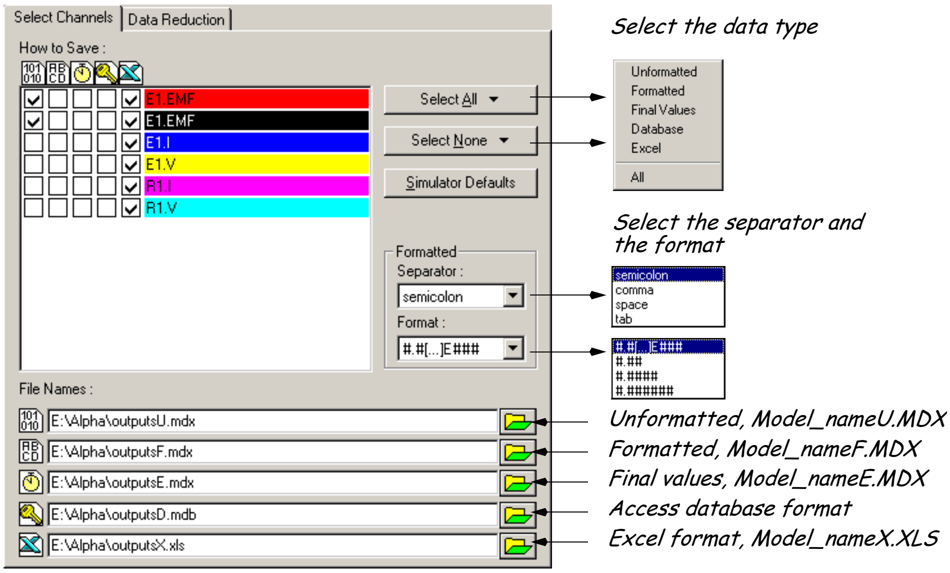
The File menu

Command	Toolbar Symbol and Description	
RELOAD LAST SIMULATION		Shows the results of last simulation. Available only under the following two conditions: 1) The **«Don't Clear Data Cache at Program End»** option is active in the OPTIONS>SETTINGS DATA. 2) The View Tool has been opened without first running a simulation (e.g., by Windows Explorer).
SAVE DATA		Saves data channels to disk in a format you select.
SAVE CONFIGURATION		Saves all window and channel settings of the active task. If a new simulation run is started, all settings of the saved configuration are applied to the new data.
PRINT		Prints the wave forms displayed on screen.
PREVIEW		Page preview. The options for print layout are specified in Options>Settings **«Print Page»**.
SELECT PRINTER		Opens the printer configuration menu.
EXIT		Closes the View Tool and stops the running simulation.

The Save Data Command

Select Channels Tab

All data channels selected for one storage format are stored together in one file. You can change the default file name and path can be changed by the user, but Ansoft recommends using the standard file extension because other SIMPLORER modules refer to these extensions.



With the buttons <Select All> and <Select None>, the file format for all available data channels can be selected or deselected at the same time.

When you click <Simulator Defaults>, the original storage options from the model file are restored.

Data Reduction Tab

Reduce extend of saving data

If this option is selected you can use the data reducing functions. The settings of these functions are applied when you save data from the View Tool to files.

Sample rate for saving

The function determines how many of the data pairs are saved. For the value "1", each data pair is saved; for "2", every other data pair; for "3", every third pair, etc.

Lower/Upper boundary of saving rang for X-axis

These options set the X-axis range for data saving. You can choose between the minimum/maximum of

- The active graphical display (e.g. a certain zoom area)
- The data provided of the simulator (simulation start and end time) or
- An own value.

Options for Storage of Simulation Data

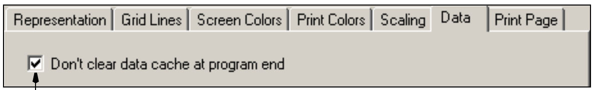
The menu OPTIONS>SETTINGS DATA provides two further options for saving simulation results. You can turn off saving simulation data to files as specified in the simulation model and keep the displayed simulation data in a additional file.

Data Tab

Don't Clear Data Cache at Program End

Online simulation data are available via the View Tool only while the View Tool is active. However, the **«Restart external View Tool»** option reopens the last View Tool used, including its simulation data. To restore the data, the option **«Do not clear data cache at the program end»** in the View Tool menu OPTIONS>SETTINGS **«Data»** must be checked. During the simulation View Tool writes the data into the temporary Windows directory. Usually these data will be deleted when you close SIMPLORER. Using the **«Do not clear...»** option, the data will remain on the hard disk and are available to be saved in a file until another simulation is started.

All simulation data (even the online outputs) will be kept. Start the View Tool in the SSC Commander and choose FILE>RELOAD LAST SIMULATION and the latest data are available immediately, without a new simulation.



View Tool data file is kept on hard disk

12.3.2 Editing Data Channels and Diagram Settings

The simulation results are presented first in the form defined in the .sml source file. With the View Tool you can change this presentation later. The edit commands for data channels are available in the EDIT menu and in the shortcut menu of a graphic. Most of them can also be selected from symbols in the toolbar.

Edit Menu ▾

Command	Toolbar Symbol and Description	
COPY		Copies the window contents to the clipboard as a bitmap.
EDIT CHANNEL		Opens a shortcut menu to edit a data channel. See also "Shortcut Menu for Data Channels" on page 389.
ADD/REMOVE CHANNELS		Opens a dialog box to select data channels.
EDIT COORDINATE SYSTEM		Defines the coordinate system settings for the coordinate system in the active graphic window. The options are available for both the X- and Y-axis. Linear or logarithmic, Scaling factors, Reset to simulator defaults. See also "Editing Diagram Settings" on page 391.
RESET		Sets the coordinate system to the original TEND and YMIN/YMAX adjustments.

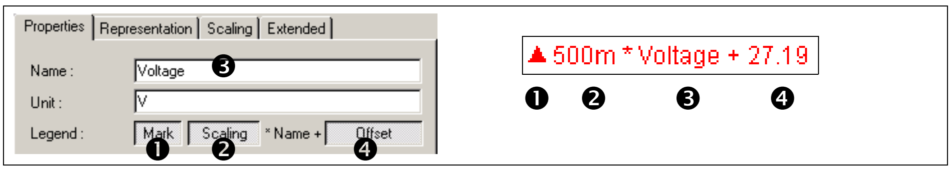
Command	Toolbar Symbol and Description
ALL CHANNELS TO SEPARATE WINDOW	Opens a separate window for each data channel.
ALL CHANNELS TO ONE WINDOW	Displays all data channels in one window (and closes all other wave form windows).
BEST REPRESENTATION	Sizes all data channels to place all signal amplitudes within one scope.

Shortcut Menu for Data Channels

Click the identifier for a data channel with the right mouse button (coordinate system legend). The menu has the following commands:

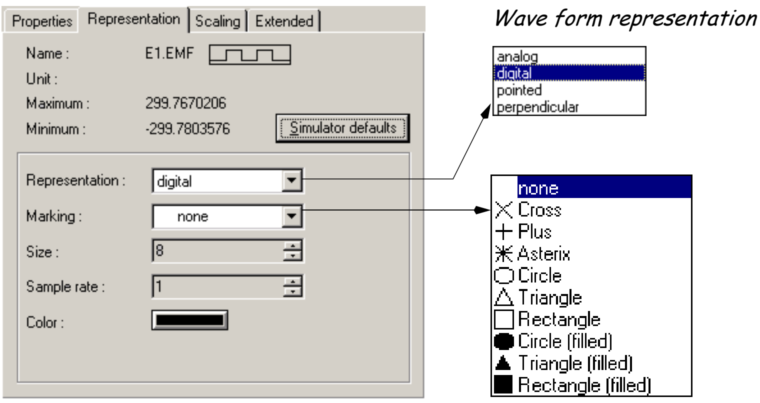
Command	Description
CURSOR	Cursor tracing of the selected wave form; numerical values appear in the status bar (bottom right).
REMOVE	Removes channel from the graphic window.
SEPARATE WINDOW	Places the selected channel in its own window.
SCALE	Sizes or shifts a channel within the diagram.
EXTENDED	Displays channel name, unit, maximum, minimum. Wave form display options can be modified (layout, line color, line marker type, etc.).
COLOR	Changes wave form color.
PROPERTIES	Display and modification of all data channel properties.

Properties Tab



Representation Tab

Using the **«Representation»** menu, markers can be used to tag a wave form.



Scale Tab

Scale and shift a selected wave form by specifying any user-defined value or using predefined factors from the button bar.

Extended Tab

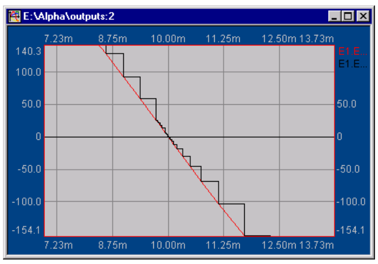
Define maximum line, minimum line, and fitness for evaluation of experiments.

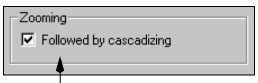
Zoom a Region

Select any part within the View Tool graphic window (drag the mouse while holding the left mouse button). The selected area is zoomed (enlarged) and displayed in a new window. The «Zooming» option in OPTIONS/SETTINGS **«Representation»** influences the size of the View Tool window after zooming.

☑ **«Following by Cascading»**
The new window with zoomed data is smaller.

☐ **«Following by Cascading»**
The new window with zoomed data retains the original size.





Zooming Option in Option Settings Representation

You can reset the zoom modification with EDIT>RESET at any time.

Editing Diagram Settings

The diagram settings are defined in the EDIT>COORDINATE SYSTEM menu. You can start the commands also with the shortcut menu of the diagram.

X Axis/Y Axis Tab

Linear or logarithmic representation and values to define minimum and maximum of the axes.

The <Simulator Defaults> function sets the axes to the values defined in the simulation model. The edit buttons apply a factor to both minimum and maximum values of the corresponding axis.

Generations Tab

Settings for generations created of experiments runs can be modified in this dialog. Please note: The option «Show» must be selected to display the generation settings.

12.3.3 View Tool Settings

The View Tool provides numerous settings for the representation of simulation results; as well as for single simulation runs and experiment runs. In addition, there are settings for the print page layout (header, footer, and margins). You can define all the settings within the OPTIONS menu of the View Tool. With <Defaults> the SIMPLORER standard values can be restored.

Options Menu ▾

Command	Description
SETTINGS	This menu includes the options for screen representation, printer adjustment and saving simulation results. See also “Options for Storage of Simulation Data” on page 387. With <Defaults> the SIMPLORER standard values can be restored.
EXTENDED	This menu includes the screen and print color options for runs in an experiment (generations).

Defining Screen Settings

This settings includes the options for screen representation, for example diagram elements, grid lines, colors, and scaling. With <Defaults> the SIMPLORER standard values can be restored.

Representation Tab

Representation
Shows or hides the coordinate system elements border, grid lines, legend, X-axis, or Y-axis.

Inscription
Shows or hides the inscription of: the X-axis (bottom or top) and the Y-axis (left or right). Sets the font for the inscription of the axis.

Graphic Border

Defines the margin width of the graphic (for the axis inscription and the legend). You can set the margin for the right, left, top and bottom margin separately.

Cursor Color

Sets the color used for the cursor. You can choose between channel color or the complementary color of the channel color.

Next Simulation

With the option «Restore Window» the View Tool window is brought to the foreground at each simulation run (refresh).

If you are doing an experiment with multiple simulations running in the background, the «Next Simulation Restore Window» option should be deactivated.

With the option «Load configuration» the saved arrangement of a former data analysis are applied to the next simulation run.

With the option «Load configuration automatically» the active arrangement is saved when the View Tool is closed. If you want to save the configuration with FILE>SAVE CONFIGURATION clear this option, otherwise the saved settings will be overwrite when the View Tool is closed.

Zooming

Determines if the windows are arranged overlapped after a zooming process.

Grid Lines Tab

Linear Representation

You can set the number of horizontal and vertical grid lines independent from each other. The defined number is used as a minimal value. This is because the View Tool tries to draw the grid lines at "round" numbers. Therefore it may be necessary to draw more lines than specified.

Logarithmic Representation

You can set the number of horizontal and vertical grid lines per decade independent from each other. If the number is 9 (grid lines at each "integer" value within the decades), additional grid lines may be drawn at "tenth" of "integer" values. These additional grid lines will occur only if the "normal" grid lines are less than a given number.

Definition of grid line number for linear or logarithmic representation. The values in the input control boxes can be modified only with the Buttons.

Screen Colors Tab

Common Colors

Click the corresponding color control box (double-click channel colors), and a dialog for color selection appears. The following elements can be set: window background, graphic background, graphic frame, X-axis, Y-axis, grid lines, X-inscription, Y-inscription.

Channel Colors

You can set arbitrarily much channel colors. Double-click an entry in the list to edit the colors. You can create new colors with the icons on the top of list. With these buttons you can also delete existing colors and change their order.

Color Scheme
In the color scheme predefined colors are listed. User specific colors can be generated and stored in new user color schemes.

The active colors set on the page can be saved as a color scheme. To save a color scheme enter a name into the input field and click <Save>. To load a color scheme select the name of the color scheme in the select box. With <Delete> a color scheme can be removed from the list. You can delete only user defined color schemes. There are some predefined color schemes: VGA Colors, Black at White, White at Black, Rainbow Colors.

Scaling Tab

X - Axis Scaling During Simulation
When a simulation is started the simulation end time is known. The simulator send the value to the View Tool which displays the corresponding size of the X-axis. If you want to see a certain part (percentage) of the complete X-axis select this option and enter the percentage. When the channels will "hit against" the right graphic border during a simulation run, the View Tool enlarges the X-axis and re-scales the channels.

Y - Axis Scaling During Simulation
The option adapts the y-axis during the simulation. When a channel get a value out of the visible range the Y-axis are scaled by the specified percentage but at least so that all points of all channels are visible.

Y - Axis Scaling at Simulation End
The option adapts the y-axis after the simulation. All channels are scaled so that all points of all channels are visible. During the simulation no scaling appears.

Defining Printer Settings

This settings includes the options for print colors and page layout. With <Defaults> the SIMPLORER standard values can be restored. The colors selected in the dialog are used for the print out. If the box «Use screen colors» is selected, print colors are identical with screen colors.

Print Colors Tab

Use Screen Colors
If this option is checked the same color scheme is used for both the screen and for printing. Black-white-printer print colors in gray. Some times small lines disappear completely. Clear this option if you have a black-white-printer.

The rest of the page
The other settings are identical with the «Screen Colors» dialog. If you have a black-white-printer select one of the predefined black-white-color-schemes or create a new scheme.

Print Page Tab

Print Border
You can set the left, right, top, and bottom margin for print in centimeter (1 inch=2.54 cm).

Header and Footer
There are available: project name, date, arbitrary text, line. Each of these elements (except the line can be placed: left, right, in the middle.

Settings for page layouts (borders, header and footer). The FILE>PREVIEW menu command shows the page layout of the active window.

Defining Extended Settings for Experiment Runs

These settings include the screen and print color options for runs in an experiment (generations, optima).

Screen Colors Tab

You can arbitrarily set generation and channel colors, as described below.

Generation Colors
Generation colors can be edited by double clicking into the list. Using the bottoms above the list, you can create new color entries, delete existing color entries, and change the order of entries.

Extended Channel Colors
You can also set arbitrarily channel colors, colors for the minimum and maximum line, and colors for the optimum range. You can edit these colors by double-clicking in the list. You can edit channel colors by clicking in the left part of the list. You can edit minimum and maximum line colors by clicking the first button in the list. You can edit the optimum range color by clicking the second button in the list. Using the buttons above the list, you can create new color entries, delete existing color entries, and change the order of entries.

The color of the optimum ranges can be determined two different way: The determination of the color of the optimum ranges can be done in different ways too. There are the options «Invert: Background» and «Invert: Color». If one of these options is checked, the optimum ranges are not displayed with the colors from the list. They are displayed with:

- the contrast color of the background or
- a color that is in contrast to the background and the channel color.

Color scheme
In the color scheme predefined colors are listed. User specific colors can be generated and stored in new user color schemes.

The active colors set on the page can be saved as a color scheme. To save a color scheme enter a name into the input field and click <Save>. To load a color scheme select the name of the color scheme in the select box. With <Delete> a color scheme can be removed from the list. You can delete only user defined color schemes. There are some predefined color schemes: VGA Colors, Black at White, White at Black, Rainbow Colors.

Print Colors Tab

Use Screen Colors
If this option is checked the same color scheme is used for both the screen and for printing. Black-white-printer print colors in gray. Some times small lines disappear completely. Clear this option if you have a black-white-printer.

The rest of the page
The other settings are identical with the «Screen Colors» dialog. If you have a black-white-printer select one of the predefined black-white-color-schemes or create a new scheme.

12.3.4 Screen Layout and Help

The View Tool provides numerous settings for arranging Window elements, the View Tool elements itself, and the single diagrams with simulation results. You can define the settings within the VIEW, WINDOW, and HELP menu of the View Tool. See also “The View Tool Main Window” on page 385.

View Menu

Command	Description
TOOLBAR	Shows or hides the toolbar.
COMMAND BAR 	Shows or hides the command bar. This option is available only if user specific commands were previously defined. These commands are initiated by clicking on the corresponding icon (letter) in the command bar or by pressing a user-defined hot key.
STATUS BAR	Shows or hides the status bar.

Arranging Windows

The View Tool can handle multiple representations of results in different windows simultaneously. The Window style and arrangement on screen can be managed through the Window menu. These commands can also be accessed directly with the icons from the toolbar.

Window Menu

Menu	Description
IDENTICAL WINDOW	Creates an exact copy of the active window. All wave forms in both the original and duplicate window are identical.
NEW WINDOW	Opens a new empty window.
WINDOW OVERLAPPING	Rearranges all open windows in an overlapping manner (cascade). All title bars are visible.
TILE HORIZONTAL	Rearranges all open windows one on top of the other.
TILE VERTICAL	Rearranges all open windows side by side.
ARRANGE ICONS	Moves all minimized windows to the lower part of the edit window and arranges them.
«LIST OF OPEN WINDOWS»	Lists all open Windows and the corresponding file names.

Help and Version Information

Online help and version information are both available with the View Tool, as for all other SIMPLORER programs, from the HELP menu.

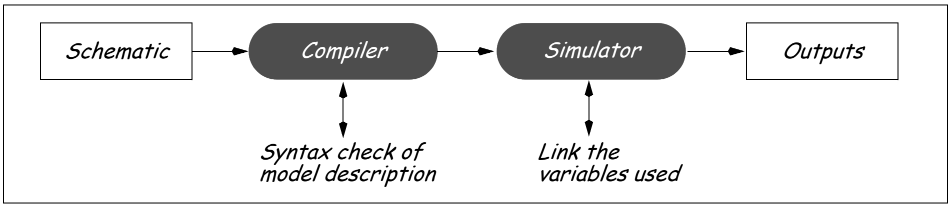
Help Menu

Command	Description
HELP CONTENTS	Go to online help.
ABOUT VIEW TOOL	Product and version information.

13 Simulator

The simulator is the heart of the SIMPLORER system. Simulation models are described in SML (SIMPLORER Modeling Language). The SML compiler, which automatically starts when you start the simulator, translates models into code the simulator can read.

At the beginning of each simulation, the simulator resolves name references of quantities used in different modules and simulators. The compiler recognizes only semantic and syntactic errors in the SML data stream. Except for the output of error messages, the compiler has no contact with and the compile process cannot be affected by the user.



SIMPLORER uses the principle of simulator coupling. In simulator coupling, different single simulators are connected to solve tasks from several technical fields and to represent their interactions. These different simulators exchange data during the simulation process, much as the real system components would exchange energy and information in a real-time physical environment.

The simulator provides a variety of features, such as user-defined simulation precision, event-driven modification of simulation parameters, and manual parameter modification during simulation.

<u>This chapter contains information on:</u>

- Simulation parameters
- VHDL-AMS simulator
- Starting simulations
- Simulator functions and facilities
- Managing the simulator queue
- Simulator backplane

13.1 Analysis Types for Simulation

The simulation model can be analyzed in three different domains:

- Time domain (transient analysis or TR simulation)
- Frequency domain (harmonic analysis or AC simulation)
- Quiescent domain (DC operating point analysis or DC simulation)
- (used to determine the operation point for models with nonlinear components).

To select a simulator, click an option from the drop-down menu on the simulation toolbar. See also “Starting the Simulator” on page 206.

Until it is changed, the last simulator selected (DC, AC, or TR) is used for future simulations. To avoid calculation errors, please check the selected simulator before calculating a new simulation model.

13.1.1 Transient Simulation

The TR Simulator is used to calculate the simulation model in the time domain. In the Schematic, choose SIMULATION>PARAMETER and click the «TR» tab to set the parameters for the TR simulator.

Components for Transient Simulation

Components for transient simulation:

- All internal components
- All VHDL-AMS components
- C models with definition for transient simulation
- Macros using appropriate models (internal components, C models, VHDL-AMS models)

Parameters of Transient Simulator

Simulation parameters control the simulation process. The choice of them are important for an successful simulation. There are general and circuit simulator parameters. The values during a simulation provide valuable information about the quality of a simulation result.

Simulation parameters can be used in equations and expressions.

General Simulation Parameters	
THETA (23)	Global ambient temperature for temperature dependent components [°C].
TEND (40m)	Simulation end time in seconds.
HMIN (10μ)	Minimum time step in seconds.
HMAX (1m)	Maximum time step in seconds.

Circuit Simulator Parameters	
SECM.SOLVER	Used integration method. 0=Euler, 1=Trapez **Trapez** Used for oscillating LC systems or analog oscillators, especially useful for investigating energy problems **Euler Integration** Used for all other simulation problems. Damping effects may cause false results for energy relationships.
SECM.LDF (0.1…10)	Local discretization error [%]. Defines the time step dependent on the dynamic of the circuit. A small error increases the precision but also the calculation time. This number itself gives no clue as to the precision of the overall result.
SECM.ITERATMAX (5…30)	Maximum number of iterations for one simulation step. If convergence problems occur, the calculation for the active simulation step will be stopped when the maximum value is reached without consideration of other error limits.
SECM.IEMAX 1µ…100µ	Maximum current sum error. A very small value lead to accurate results, but convergence problems might be possible in the Newton integration method.
SECM.VEMAX 1m…1	Maximum voltage error. A very small value leads to accurate results, but convergence problems may occur using the Newton integration method.

Basic Rules for the Proper Choice of Time Step

Correct simulation processing and results depend on the proper choice of minimum and maximum values for the integration step size. The smaller the maximum integration step size, the more correct the results, but the longer the processing time.

This means that, when specifying these minimum and maximum time step values, you need to compromise between accuracy and time. The basic rule of measurement “Not as precise as possible, but as precise as required” is also valid for a simulation. The following guidelines should help you prevent elementary mistakes in choosing the proper integration step width:

Model properties	Recommended
What is the smallest time constant (τ_{min}) of the electric circuit (R*C or L/R) or of the block diagram (PTn-elements)	$Hmin < \frac{\tau_{min}}{10}$
What is the largest time constant (τ_{max}) of the electric circuit (R*C or L/R) or of the block diagram (PTn-elements)	$Hmax < \frac{\tau_{max}}{10}$
Which is the smallest cycle (T_{min}) of oscillations that can be expected (natural frequencies of the system or oscillating time functions)	$Hmin < \frac{T_{min}}{20}$
Which is the largest cycle (T_{max}) of oscillations that can be expected (natural frequencies of the system or oscillating time functions)	$Hmax < \frac{T_{max}}{20}$

Model properties	Recommended
What is the smallest controller sampling (TS_{min})	$Hmin < \frac{TS_{min}}{5}$ $Hmax = TS_{min}$
What is the fastest transient occurrence (TU_{min}) (edge changes of time functions)	$Hmin < \frac{TU_{min}}{20}$
What is the time interval to be simulated (Tend)	$Hmax < \frac{TEND}{50}$

- Select the smallest of each estimated maximum and minimum time step for your simulation model.
- All the values recommended above are based on numeric requirements and experience and do not guarantee a successful simulation. Please consider the algorithm as a guideline.
- In case of doubt, decrease the maximum and minimum step size by dividing by 10, repeat the simulation and compare the results. If the second set of results (with the step size decreased) shows conformity with the first results, then the step sizes chosen for the first simulation were appropriate (remember that smaller values increase the simulation time).

If the number of iterations is identical with the defined maximal value (NEWTON) during the simulation, the model may be incorrect. The simulation monitor displays the active number of iterations. See also 13.4.1 “Displaying Monitors” on page 412.

13.1.2 AC Simulation

The AC Simulator is used to calculate the simulation model in the frequency domain. First it performs a DC simulation to calculate operation point values and then an AC simulation for a given frequency range. The voltage and current information for the DC simulation and the values for the AC simulation are saved in one .sdb file. If you open the file in the DAY Post Processor you can choose between the two results To display gain and phase diagram use the Bode Plot Element. See also “Bode Plot” on page 377.

Components for AC Simulation

Components for AC simulation in the «Basics» tab:

- Passive Components
- Electrical Sources (except for Fourier source)
- Switches (except for controlled switches)
- Semiconductor System Level
- Semiconductor Device Level
- Spice compatible models
- Transformers
- Continuous Blocks
- Discrete Blocks

- Source Blocks
- Signal Processing Blocks
  (except for MAX, MIN, MAXT, MINT, two-point element with hysteresis)
- Math Blocks
- Measurement (Electrical domain)
- Time Functions
- Characteristics
- Equations (except for DES solver)
- C models with definition for DC and AC simulation.
- Macros using appropriate models (internal components, C models)

Not supported components for AC simulation in the «Basics» tab:

- Electrical machines
- State graph components
- Signal Characteristics
- Physical Domain Components

Not supported components for AC simulation in the other tabs:

- Components in the «AMS» tab
- Components in the «Digital» tab
- Components in the «Tools» tab

If models without DC and AC implementation are used in an AC simulation, an error message appears. To leave components without DC and AC implementation on the sheet, select the components and choose ELEMENT>DON'T ADD TO MODEL DESCRIPTION. When you want to process a transient simulation, you can change this option.

AC Simulation Models

In addition to invalid network configurations, the following configurations cannot be used with AC simulations:

- Parallel connection of voltage sources and inductances
- Parallel connection of current sources and capacitances without any other branch

See also “Network Configurations” on page 71.

If an AC simulation is started, the values for magnitude and phase or real and imaginary part, defined in electrical and block sources, are used to generated the sine wave.

The magnitude of AC parameters can be a numerical value, a variable (defined in an initial assignment condition or equation), or an expression. If you use an expression, only quantity types with the attribute Out and standard functions can be used, for example R10.V*ABS(C1.V). In contrast to the numerical value and the variable, expressions will be linearized in the operation point. Therefore, expressions also contain information about the current phase whereas numerical values and variables have no phase information.

The parameters phase, real part, and imaginary part are common parameter types. You can use all numerical values, varibales, or expressions.

Parameters of AC Simulator

In the Schematic, choose SIMULATION>PARAMETERS and click the «AC» tab to set the parameters for the AC simulator.

Voltage and current for AC analyses are defined for electrical sources and block sources in the simulation model. In the «AC - Parameters» dialog of sources you can define either magnitude and phase or real and imaginary part of the AC value. This value can be constant or dependent on other quantities. In the View Tool and the Display Elements, wave forms are drawn in the frequency domain. See also 12.2.1 "View Elements" on page 369.

General Simulation Parameters	
`THETA` (23)	Global ambient temperature for temperature dependent components [°C].
`FSTART` (40m)	Start frequency [Hz].
`FSTOP` (10µ)	Stop frequency [Hz].
`FSTEP` (1m)	Frequency step width (linear sweeptype) or points per decade (decadic sweeptype).
`ACSWEEPTYPE` (1)	Sweep type, 0=linear or 1=decadic.

Solver Parameters	
`EMAXAC` (1u)	Maximum error [A]

Selecting AC Simulator for Simulation

13

To start an AC simulation the AC simulator must be selected in the simulation toolbar. Open the drop-down list and select **«Simulator AC»**. If you start a simulation, the simulator selected in the toolbar, is used. Please note, the simulator is used for all following simulations as long as another simulator is selected.

13.1.3 DC Simulation

The DC Simulator is used to calculate to determine the operation point for simulation models with nonlinear components in the quiescent domain. The voltage and current information about the operating point is saved to the .sdb file of the simulation model automatically. The voltage values for the computed operation point is automatically displayed on the sheet at the components. The menu commands CONNECT>SHOW DC RESULTS and CONNECT>HIDE DC RESULTS switches the labels at the components on and off. See also "Connect Toolbar and Menu" on page 48.

Components for DC Simulation

Components for DC simulation in the «Basics» tab:

- Passive Components
- Electrical Sources (except for Fourier source)
- Switches (except for controlled switches)
- Semiconductor System Level
- Semiconductor Device Level

- Spice compatible models
- Transformers
- Continuous Blocks
- Discrete Blocks
- Source Blocks
- Signal Processing Blocks
  (except for MAX, MIN, MAXT, MINT, two-point element with hysteresis)
- Math Blocks
- Measurement (Electrical domain)
- Time Functions
- Characteristics
- Equations (except for DES solver)
- C models with definition for DC simulation.
- VHDL-AMS models with definition for DC simulation.
- Macros using appropriate models (internal components, C models)

Not supported components for DC simulation in the «Basics» tab:
- Electrical machines
- State graph components
- Signal Characteristics
- Physical Domain Components

Not supported components for DC simulation in the other tabs:
- Components in the «AMS» tab
- Components in the «Digital» tab
- Components in the «Tools» tab

If models without DC implementation are used in an DC simulation, an error message appears. To leave components without DC implementation on the sheet, select the components and choose ELEMENT>DON'T ADD TO MODEL DESCRIPTION. When you want to process a transient simulation, you can change this option.

DC Simulation Models

In addition to invalid network configurations, the following configurations cannot be used with DC simulations:

- Parallel connection of voltage sources and inductances
- Parallel connection of current sources and capacitances without any other branch

See also "Network Configurations" on page 71.

Parameters of DC Simulator

In the Schematic, choose SIMULATION>PARAMETER and click the «DC» tab to set the parameters for the DC simulator.

Solver Parameters	
THETA (23)	Global ambient temperature for temperature dependent components [°C].

Solver Parameters	
ITERATMAX (50)	Maximum number of iterations.
EMAXDC (1m)	Maximum error [A].
RELAXMAX (10)	Maximum number of relaxations.

13.2 VHDL-AMS Simulation

The VHDL-AMS simulator is a sub-simulator of the SIMPLORER system. It calculates simulation models described in VHDL-AMS (Very high speed integrated circuit Hardware Description Language - Analog Mixed Signal). The SML compiler starts the VHDL-AMS simulator if VHDL-AMS models are used in the simulation model.

Components for VHDL-AMS Simulation

- Components in the «AMS» tab.
- Components in the «Digital» tab.
- Components with implementation of VHDL-AMS description.

If a macro models has both, SML and VHDL-AMS description, you can define the used modelling language in the component dialog. Double-click the component and click the «Library» tab. Open the drop-down list **«Select modelling language»** and select either SML or VHDL-AMS. If VHDL-AMS is selected, you can also choose an architecture in the drop-down list **«Select Architecture/Modelling Level»**. See also “Creating a Model from Existing .vhd File” on page 490 and “Creating Models in VHDL-AMS in the Model Agent” on page 492. See also VHDL-AMS model description in the SMPLORER online help.

Parameters of VHDL-AMS Simulator

In the Schematic, choose SIMULATION>PARAMETER and click the **«TR»** tab to set the parameters for the VHDL-AMS simulator. See also 13.6 “Simulator Backplane” on page 418.

General Simulation Parameters	
THETA (23)	Global ambient temperature for temperature dependent components [°C].
TEND (40m)	Simulation end time in seconds.
HMIN (10µ)	Minimum time step in seconds.
HMAX (1m)	Maximum time step in seconds.

Circuit Simulator Parameters	
SECM.SOLVER	Used integration method. 0=Euler, 1=Trapez **Trapez** Used for oscillating LC systems or analog oscillators, especially useful for investigating energy problems . **Euler Integration** Used for all other simulation problems. Damping effects may cause false results for energy relationships.

Circuit Simulator Parameters	
SECM.LDF (0.1…10)	Local discretization error [%]. Defines the time step dependent on the dynamic of the circuit. A small error increases the precision but also the calculation time. This number itself gives no clue as to the precision of the overall result.
SECM.ITERATMAX (5…30)	Maximum number of iterations for one simulation step. If convergence problems occur, the calculation for the active simulation step will be stopped when the maximum value is reached without consideration of other error limits.
SECM.IEMAX 1µ…100µ	Maximum current sum error. A very small value lead to accurate results, but convergence problems might be possible in the Newton integration method.
SECM.VEMAX 1m…1	Maximum voltage error. A very small value leads to accurate results, but convergence problems may occur using the Newton integration method.

Differences to Standard VHDL-AMS

SIMPLORER Schematic

- Model parameter in SIMPLORER style (no TIME, 'LEFT,…)
- Vector elements in the property box of the model ({0,1,0,0}, "01010", 1=>5.0, 3=>4.0, other=>0.0) cannot be initialized.
- No signals in SIMPLORER SML, therefore all SIMPLORER quantities creating events like a signal.
- No difference between ports and generics in SIMPLORER.
- No display of physical units (sec, hour, …) in a Display Elements.
- Physical values in base units (Time in fs)
- No C Interface.

Structures

- No port and generic map between blocks.

Data types

- No pointers.
- No composite natures.
- No quantity arrays simultaneous statements.

Unsupported VHDL statements

- Generate
- Alias, Group
- User-defined attributes
- RESOLUTION function
- LIMIT/TOLERANCE groups

Unsupported attributes

- Q'Tolerance
- DT'Range
- Terminal attributes

13.3 Starting the Simulator

To compute a simulation model, the simulator can be started from four different SIMPLORER programs:

- Schematic (active simulation model—.ssh file)
- SSC Commander (simulation model in SML—.sml file)
- Experiment tool (simulation model in SML—.sml file)
- SIMPLORER text editor (active simulation model—.sml file)

13.3.1 Simulation in Schematic

When you have completed designing the simulation model, by placing, connecting, and defining all components, you can then process the simulation.

You need to assign outputs to have simulation results in Display Elements, the View Tool, or files. Appropriate simulation parameters are required for good results. Please check the outputs and parameters before you start the simulation.

The simulation parameters are accessed through the «TR», «AC», «DC», and «General» dialogs in the SIMULATION>PARAMTERS menu. The X and Y range of the View Tool are defined in the «View Tool» dialog of SIMULATION>OUTPUTS.

Simulating Online

The simulator, defined in the simulation toolbar (TR, AC, or DC), can be started with:

- the F12 key
- the button on the toolbar
- the menu SIMULATION>START

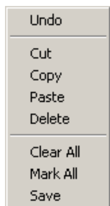
When the simulator is started, the model is compiled and the simulation starts. Every time the simulator is started, the active file is saved.

During the simulation run, the file name of the model is visible in the simulation toolbar, and the symbols to stop and break the simulation are available. At the end of simulation, the program remains open, so that the simulation can be continued with SIMULATION>CONTINUE to a new simulation end time.

From the simulated file a backup file is created. To save the file including online outputs, choose FILE>SAVE after the simulation run.

The Information Window

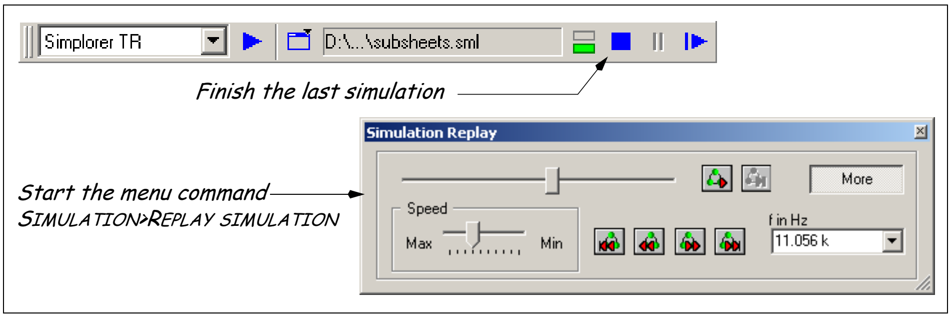
The Information Window displays error messages from the compiler and the simulator. These messages can help the user to eliminate errors in the model description. A shortcut menu can be opened with a right mouse click in the information window. With the commands in this menu you can copy messages into the clipboard and insert them into a text editor. Whenever an error is found, the Information Window is automatically displayed with the error message.



13

Simulation Replay Function

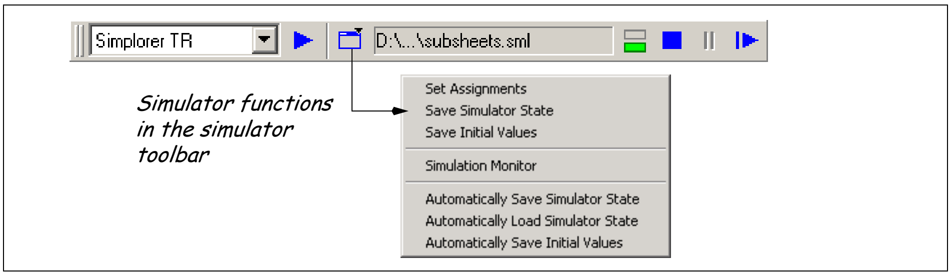
With the Simulation replay function you can play simulation runs several times, at different speeds, step-by-step, or continuously, without computing the model again. To use the replay function, the simulation results must be available in Display Elements and the last simulation must have finished **completely** with SIMULATION>STOP. The SIMULATION>REPLAY SIMULATION menu command opens the dialog to start an offline simulation.



All outputs in Display Elements and the states of animated symbols are displayed on the sheet during an offline simulation. The replay function is very helpful for finding errors when the simulation run must be checked step by step. In the replay mode, animated symbols are kept in the last state of simulation. If you close the dialog, the symols are set to their start state.

Using Simulator Functions

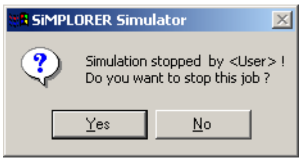
Simulator functions are available in the simulation toolbar menu, in the Schematic SIMULATION menu, and in the simulator icon menu in the Windows taskbar. See also 13.4 “Simulator Functions” on page 411.



Schematic Simulation Menu

Command	Toolbar Symbol and Description
PARAMETERS	Opens the PROPERTY menu to define simulation parameters.
OUTPUTS	Opens the sheet output dialog. All sheet outputs, depending on the options in the dialog, are listed. You can delete as well as add new outputs.

Command	Toolbar Symbol and Description
Start	Starts a simulation and compiles the .sml source file, which is created from the Schematic sheet at simulation start.
Model Description	Creates a separate .sml or .vhd file from the simulation model (without starting a simulation).
Stop	Completely stops a running simulation.



Break	Temporarily breaks (pauses) a running simulation.
Continue	Resumes a paused simulation.
DAY Post Processor	Starts the DAY Post Processor. If simulation results (.sdb file) exist for the active model, this file is opened in a new window. To save online outputs in .sdb files, choose File>Save after a simulation run.
Experiment Tool	Starts the Experiment tool.
Restore Animated Symbols	Initializes animated symbols. (Markings in states, states of switches).
Replay Simulation	Replays the entire simulation process using the simulation results in Display Elements and Animated Symbols (without recomputation).

13

The Simulator Queue

Normally the simulation model is computed immediately after the simulator call. If there are other active simulation tasks, the new task are placed at the end of the queue. If you wish, you can change the process sequence in the SSC Commander. See also 13.5 “Managing the Simulator Queue” on page 416.

Model Description

The Simulation>Model Description command creates an source files from the active simulation model in the Schematic (without starting a simulation).

To create a description (SML or VHDL-AMS) from an existing graphical model start the Simulation>Model Description command. After entering the file name and selecting the format, the description is created. Subsheets in the model description are imported as macro definitions. From model descriptions of this form, the Model Agent can create macros which can be used in any other simulation model.

13.3.2 Simulation in the SSC Commander

The simulator can be started directly from the SSC Commander by selecting an .sml file prepared for simulation. The SSC Commander also manages the simulator queue, which contains the simulation tasks ready for processing.

Once an .sml file from the project file list is selected, the simulator, defined in the simulation toolbar (TR, AC, or DC), can be started with:

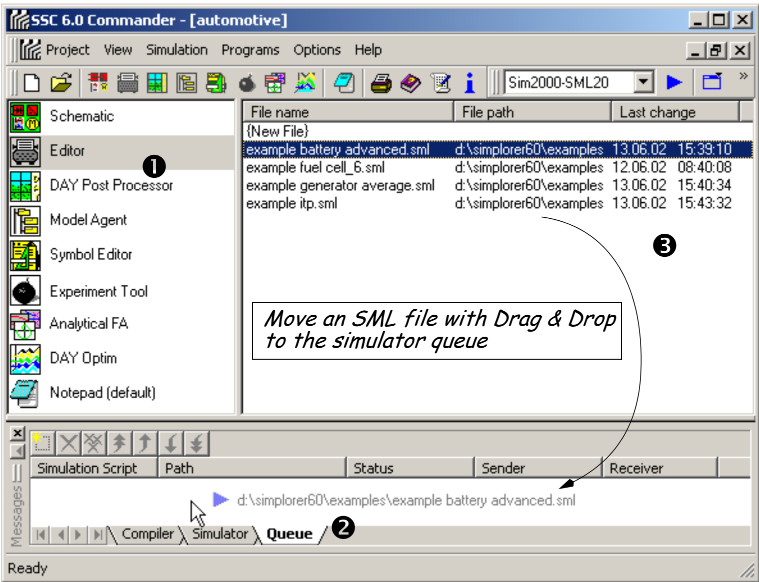
- the F12 key,
- the command SIMULATION>START, or
- the symbol on the simulation toolbar.

While the simulation is running, the model file name is visible in the display box. The symbols for stopping and breaking (pausing) the simulation are active.

From the SSC Commander, the simulator can be started only for model files already existing in SML. Schematic files (.ssh) cannot be opened directly with the SSC (.ssh files can be opened using the SIMPLORER Schematic).

Starting a Simulation with Drag-and Drop

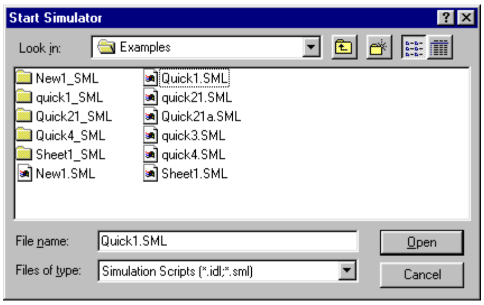
You can also start the simulator by dragging a simulation model in SML from the project list into the simulator queue. This feature requires a project with at least on SML description. Select the Editor in the left pane and the «Queue» tab in the information window. If an .sml file is represented in the list you can drag it into the information window. The View Tool is opened if online outputs were defined in the simulation model.



Starting a Simulation with Dialog Selection

Choose SIMULATION>SIMULATE MODEL or click the symbol and select an existing .sml file in the dialog

The selected .sml file is added to the simulation queue and processed. The file is **not** included in the project.



Selection of a simulation source file

Starting and Controlling a Simulation with Menu Commands and Functions

To start the simulator for the selected model, you can use the commands on the SIMULATION menu or the symbol in the simulation toolbar. Please note: The simulation menu in the SSC Commander is only available with a opened project.

SSC Commander Simulation Menu

Command	Toolbar Symbol and Description
START	Starts a simulation and compiles the selected .sml source file.
STOP	Completely stops a running simulation.
BREAK	Temporarily breaks (pauses) a running simulation.
CONTINUE	Resumes a paused simulation.
SIMULATE MODEL	Selects an existing .sml source file for simulation.

13.3.3 Simulation in the Text Editor

When you start the simulator from the text editor, the active simulation model will be calculated. The simulator, defined in the simulation toolbar (TR, AC, or DC), can be started with:

- the F12 key,
- the command SIMULATION>START, or
- the symbol on the simulation toolbar.

Text Editor Simulation Menu

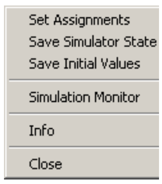
Command	Toolbar Symbol and Description
START	Starts a simulation and compiles the .sml source file
STOP	Completely stops a running simulation.
BREAK	Temporarily breaks (pauses) a running simulation.
CONTINUE	Resumes a paused simulation.

13.3.4 Simulation in the Experiment Tool

With <Start> the simulator performs multiple simulations of the model with the options and parameter variations specified within an experiment. In addition to the simulation monitor, the Experiment tool monitor displays the processing status and active parameter settings.

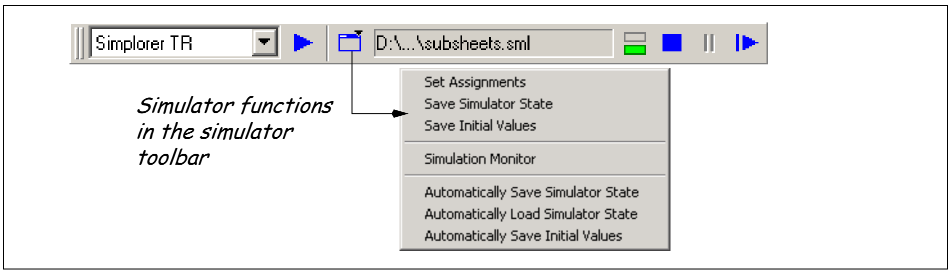
13.4 Simulator Functions

After starting a simulation, the simulator menu offers different functions to displays and control the simulation process. The menu can be brought to the foreground by clicking on the simulator kernel icon in the Windows task bar.



Please note: The simulator kernel icon is invisible if the Windows taskbar is hidden. In this case move the mouse pointer to the screen edge where the taskbar is hidden. The Windows taskbar slides into view.

In addition to the icon in the Windows task bar, the commands on the SIMULATION toolbar can also be used.



Simulator functions in the simulator toolbar

13.4.1 Displaying Monitors

After starting a simulation, you can display the simulator state monitor. The monitor offer a variety of information about the simulation's progress, especially if graphic outputs in the View Tool or Display Elements are not being used, to reduce calculation time.

Choose **«Simulation Monitor»** on the simulator. Start the simulator menu either with the Windows taskbar icon or with the simulation toolbar command).

Displaying the Simulator State Monitor

The simulator state monitor shows the name of the simulation source file being processed, the calculation progress, the start and stop time of simulation, and the duration of the simulation, as well as active simulation parameters. The dialog text box shows those outputs generated by the DIS and TXT functions (actions) of the user's state graph.

Click the icons and , and the simulation can be stopped temporarily and resumed.

Displaying the Circuit Simulator Monitor

The monitor for the circuit simulator is available only when circuit components belong to the simulation model. The circuit simulator monitor shows the simulation parameter of circuit simulator and the curve of iterations during the simulation. The active number of iterations is an important feature to evaluate simulation results. The number should be clear lower than the defined maximum for iterations. When the number of iterations is a time at the upper limit, decrease the simulation time step or increase the number of iterations.

13.4.2 Using Simulator Files

The simulator is able to create two files with data at a certain moment of simulation:

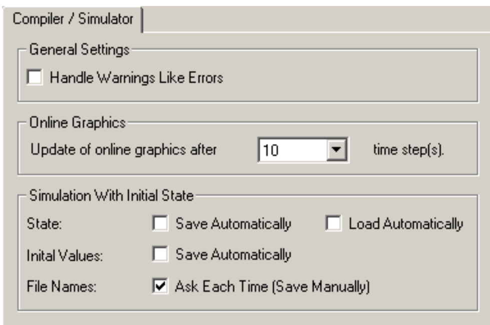
- A simulator state file with all values at a specific simulation time (.krn)
- A file with the active values of capacitors and inductor (.aws)

Both files can be used to start a simulation with predefined quantities for initial values of capacitors and inductors (initial values) or for the complete simulation model (simulator state). If one of these files is created at a time a message is written in the Information window.

Settings in the SSC Commander for Simulator Files

To simulate with initial values and state files, the corresponding files must be created after the simulation. These files can be created manually or automatically.

The handling of state or initial value files depends on the settings made in the SSC Commander OPTIONS>SIMULATOR SETTINGS or in the Simulator Toolbar menu.



Dialog box simulator settings

State

With the state file you can continue a paused simulation later. The saving options can be set in the SSC Commander OPTIONS>SIMULATOR SETTINGS or the Simulator Toolbar menu.

☑ **Save automatically:**
The simulator state is saved automatically at the end of each simulation run (.krn file). Any existing state file is overwritten.

☑ **Load automatically:**
When starting a simulation model, if there is a corresponding state file in the model directory, a dialog box for simulation start with the existing state file opens automatically.

Initial Values

With the initial value file (.aws) you can start a simulation with specific initial values for the following components:

- Capacitors (initial voltage)
- Inductors (initial current)
- Electrical Machines (initial speed, position, currents)
- Integrator Block
- Derivative Block
- Two-point Element with Hysteresis
- Dead Time Block
- Sample and Hold Block
- States in State Graphs

☑ **Save automatically:**
Values of capacitors and inductors are saved automatically at the end of each simulation run (.aws file).

The menu commands SAVE SIMULATOR STATE and SAVE INITIAL VALUES in the simulation toolbar can also write simulation data from a paused simulation.

Using Simulation State

1 Before you can simulate with a state file (.krn), the state file must be created from a previous simulation of the same model. The state file contains the values of all system quantities at a specific simulation time. To create a state file, you have the following options:

- Save state manually
  If you start the SAVE SIMULATOR STATE command on the simulator menu, a state file *Model_name.krn* is created.

  You can change this default file name. Please note: The simulator starts the dialog for using simulator state only when a state file exist, which has the same name as the model file and the file is located in the model directory.
- Save state automatically
  If you select the «State: Save automatically» option in the SSC Commander OPTIONS>SIMULATOR SETTINGS or «Automatically Save Simulation State» on the Simulator Toolbar menu, the state file is saved automatically for each simulation run.

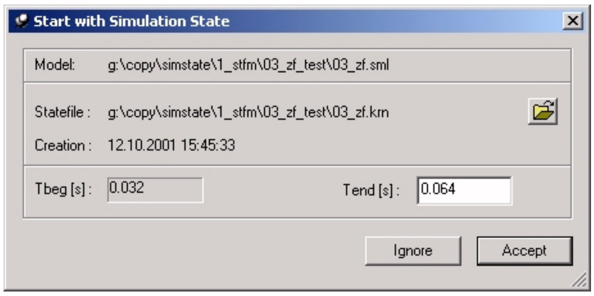
2 How to use a state file

Conditions:
When you start the simulator, the «Start with Simulation State» dialog opens under the following conditions:

- The state file is located in the model directory of the simulation model and has the same name as the simulation model (*Model_name.krn*).
- The simulation model has not changed since the last saving of the state file.
- The «State: load automatically» option in OPTIONS>SIMULATOR SETTINGS of the SSC Commander or on the Simulator Toolbar must be checked.

If these conditions are met, the dialog «Start with Simulation State» opens when you start the simulator.

You can open another state file by clicking the symbol.



State file options for simulation start

Model:
Name of model file (SML source file).

State file:
Name of state file used for simulation with the stored simulation state.

Created:
Date of state file creation.

Tbeg:/Tend:
Simulation start time [s]./Simulation end time [s].

3 Define the new simulation end time and start the simulation with <Accept>. If you click <Ignore>, the values from the .sml source file are used.

The new simulation end time `Tend` must be greater than the start time Tbeg. If not, when you click <Accept>, you will see an error message.

Using Initial Values

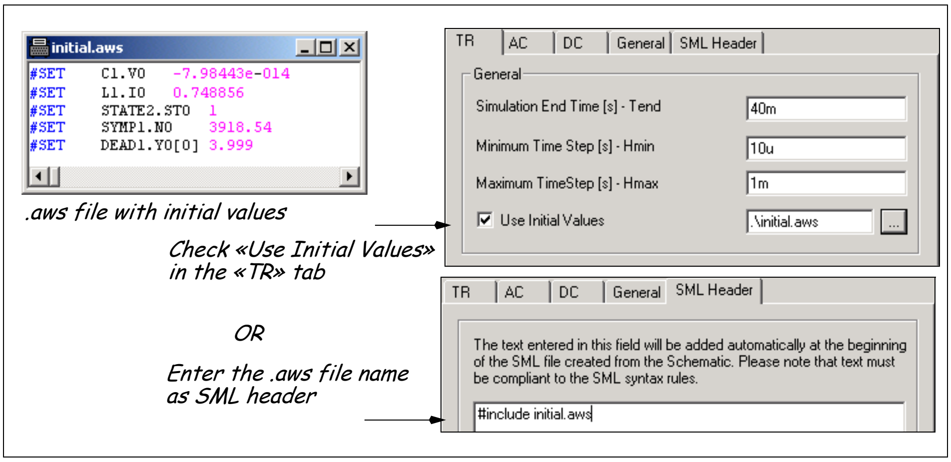
1 Save initial values

- Save file manually
  The SAVE INITIAL VALUES command in the simulator menu creates the file *Model_name.aws*. You can change the file name in the Windows file save dialog. Please note that this file must be placed in the model directory.
- Save state automatically
  If the «Automatically Save Initial Values» option in OPTIONS SIMULATOR SETTINGS of the the SSC Commander or the Simulator Toolbar is checked, the file with initial values is saved automatically.

2 Include an .aws file in the simulation model.

To include an existing aws.file you can choose one of these variants:

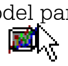
- Insert as Simulator Parameter
  You can include the created the initial value file within the «TR» dialog. Choose SIMULATION>PARAMETERS to open the dialog and check the box «Use Initial Values». If you want, you can change the default file name.
- Insert as Header
  You can include the aws. file in the SML header within the «SML Header» dialog. Choose SIMULATION>PARAMETERS and enter the statement *#include name.aws* in the edit field. The file must be located in the directory of the simulation model). See also "Initial Values and Simulation State" on page 54.



13.4.3 Other Simulator Functions

In addition to the state and initial value files of the simulator there are the simulation function to set simulation quantities during the simulation. In the simulator menu you can start the feature with SET ASSIGNMENTS. You can also change simulation parameters.

Modifying Parameters during a Simulation

With the SET ASSIGNMENTS function, simulation and model parameters can be influenced while a simulation is running. Click the simulator icon  and choose **«Set Assignments»** to open the dialog.

Simulator Tree

Selection of the simulator module (time functions, electrical circuit, integration parameter, and so on) of which model or simulation parameters are to be modified.

Parameters Folder
List of simulation parameters (TEND, HMIN, HMAX).

Models Folder
List of all components used in the simulation model and their parameters.

SECM
List of circuit simulator parameters and a list of all components used in the circuit simulator, and their parameters.

SBDM
List of all components used in the block simulator, and their parameters.

SSGM
List of all components used in the state graph simulator, and their parameters.

Value
Text box to enter a new value for the selected parameter.

<Set>
Accept the modifications.

<Close>
Close the dialog box.

Click the symbols and to stop temporarily and resume. The influence of any modified parameters can thus be tested immediately.

If the sub menu «Set Assignments» does not appear in the simulator menu, the dialog is already open but may be covered by other applications. Use ALT+TAB to switch between the applications to the desired window.

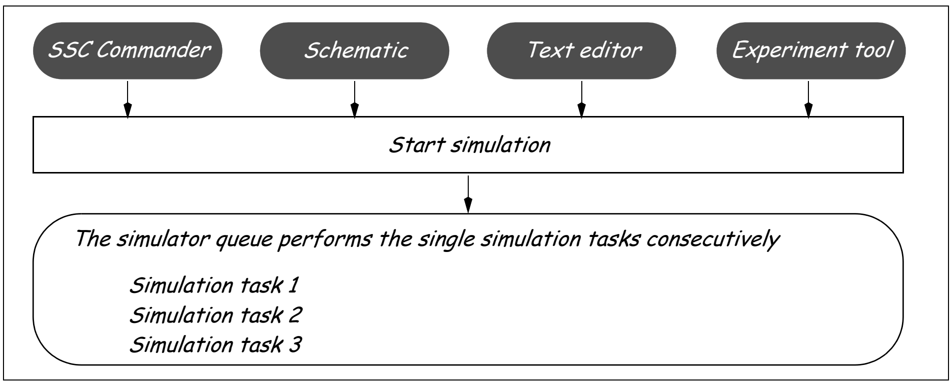
Simulator Function Close

Closes the simulator and clears all simulation tasks from the simulator queue. If you close the View Tool, any running simulation is stopped.

The commands SHEET>SYNCHRONIZE and SIMULATION>REPLAY SIMULATION are available only if the last simulation task was closed («Close»).

13.5 Managing the Simulator Queue

When the simulator is started in a SIMPLORER program, the simulation task is added to the simulator queue. You can manage the simulator queue in the SSC Commander.

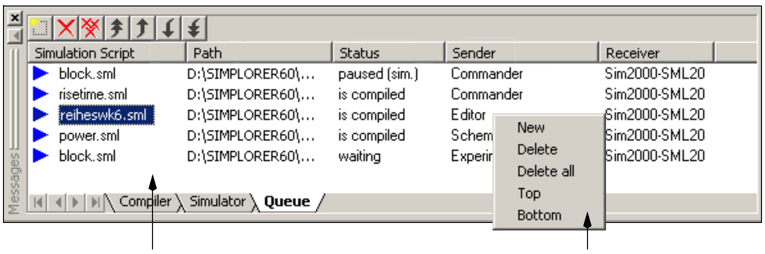


Shortcut Menu and Simulator Queue Icons

Shortcut Menu	Icon	Description
New		Load an existing simulation file and insert it in the simulation queue.
Remove		Delete a selected simulation task.
Remove all		Delete all simulation tasks in the queue.
Top		Move a selected simulation task to the top of the queue (first calculated).
		Move a selected simulation task up one position in the queue.
Bottom		Move a selected simulation task to the bottom of the queue (last calculated).
		Move a selected simulation task down one position in the queue.

Changing the Simulation Sequence

To change the processing sequence of simulation tasks, select an entry and click one of the symbols on the upper left side of the information window, or choose a command from the shortcut menu (click the model name with the right mouse button).

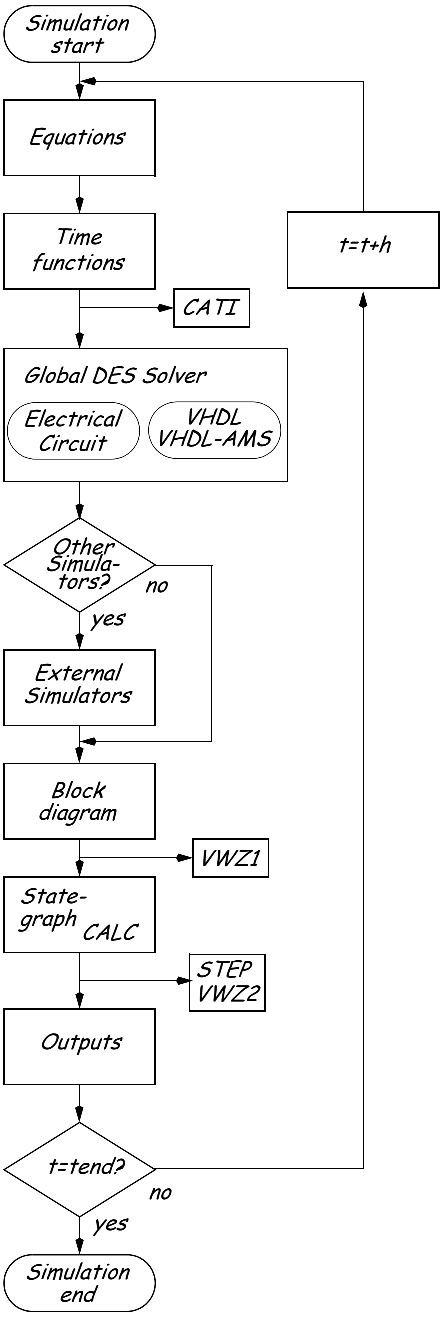


Simulator queue in the SSC Commander

Shortcut menu

If the first simulation in the queue will not start, close the simulator by clicking icon on the Windows taskbar. The next simulation task starts automatically.

13.6 Simulator Backplane



The principle of simulator coupling is often described as simulation backplane technology, because simulators are coupled by a data bus performing the communication between the simulators.

Because certain basic knowledge about the simulation methods applied here is important for successful processing of a simulation task, below is a brief review of the fundamentals of SIMPLORER's numerical algorithms and simulator coupling.

Although in reality all processes of a complex technical system run in parallel, in simulation a sequential calculation of these processes must be done, because most readily available computers have only one processor. Therefore, one must always keep in mind during modeling that all the components of a modeled system are processed one after another.

Circuit Simulator Processing

The circuit simulator is based on a modified nodal approach. For the solution of the differential equation system, a numerical integration method (either the Euler or the trapezoidal algorithm) is applied. The solution of nonlinear equations is done by the Newton-Raphson method; the calculation of equation systems, linearized in the operating point, takes place by means of LU factorization after Gauss.

An important feature of the applied solvers is the automatic time step control. This feature ensures an adaptation of the calculation step width dependent on the active dynamic situation in the system considered, in such a way that there is always an optimum between precision of calculation and simulation speed. The time step limitations are user-defined by a minimum and maximum time step. Through this same specification, the possible time step range for the active simulation task is also determined:

`TEND`:	Simulation duration
`HMIN`:	Minimum time step width permissible
`HMAX`:	Maximum time step width permissible

The limits for number of time steps are, consequently:

$$max_{step} = \frac{t_{end}}{h_{min}} \quad \text{and} \quad min_{step} = \frac{t_{end}}{h_{max}}$$

The real number of time steps would be between these two boundaries. If the step width for a specific time step is too large (automatically recognized), the circuit simulator requests a cancellation of this time step, and it will be repeated with an adapted step width (also calculated automatically). The necessary step cancellations are forwarded to a step width manager and processed there.

Block Diagram Simulator Processing

Block diagrams are calculated according to the block transfer characteristics and in the sequence of signal flow. Integrators are processed with the Euler method and according to the principle of distributed integration.

For each block a certain sampling time can be defined by the user. Then the blocks are equidistantly calculated only on these discrete times. The sampling time must be selected very carefully. There may be dead time effects, especially in simulation models with different sampling times.

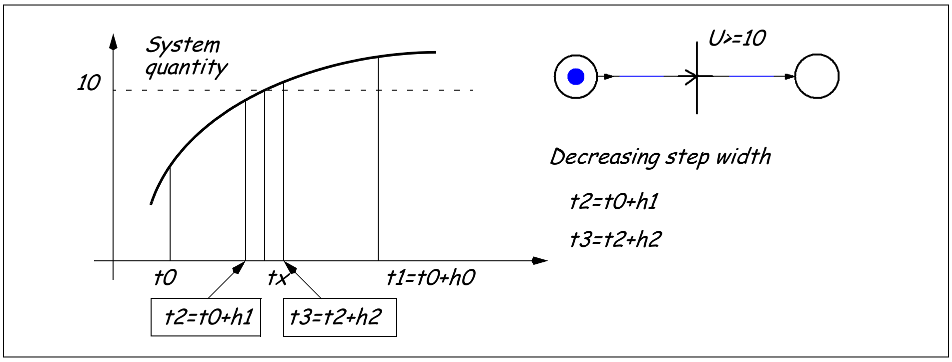
If no sampling time is specified, the block diagram module runs with the same (variable) time step as the circuit simulator (quasi continuous case). In this case, make sure that the block diagram time constants are larger than those in the electrical circuit model. If no sampling time is specified and there is also no electrical circuit in the model, the block diagram module runs constantly with `HMAX`.

The observance of block diagram simulator calculation times is guaranteed by the predictive time step control of the time step manager.

State Graph Simulator Processing

Within state machines the calculation of states, transitions, and actions continue as long as there is no other valid transfer condition. The calculations are executed only in active (marked) states. Afterwards the system tests if there have been any events. Only if events have occurred and further states can be marked are new actions calculated.

State graphs do not need their own time step control; they work in discrete time and are actualized at each time step. Otherwise, the state graph simulator itself influences the time step (in the same way as other SIMPLORER modules). This method of processing concerns first of all the identification of events. For the time determination of such an event, the simulator searches for a value as precise as possible, canceling and repeating simulation time steps with a decreasing time step until the minimum time step is reached. At this limit the time determination of an event is accepted as valid (with sufficient precision) and the simulation continues as usual. This event synchronization is performed using ≥ or ≤ and bypassed on the > and < operators.



14 Data Evaluation

The basic version of SIMPLORER comes with powerful visualization tools. Simulation results can be analyzed and visualized online via Display Elements and the View Tool. With the DAY Post Processor and DAY Optim programs, you can analyze results using mathematical functions, and process and rearrange data in spreadsheets. Using the DAY Post Processor's presentation mode, you can create and print graphics, tables, and pictures.

<u>This chapter contains information on</u>

- DAY Post Processor to evaluate simulation results
- DAY Optim to evaluate experiments
- File formats

14.1 DAY Post Processor

The DAY Post Processor program allows you to conduct detailed graphical and numerical data analyses. DAY analysis files (with extension .day) can contain several simulation results files, including user-defined graphical representations. To start the DAY Post Processor from the SSC Commander, double-click the DAY Post Processor symbol or choose PROGRAMS>DAY POST PROCESSOR. You can also choose SIMULATION>DAY POST PROCESSOR in the Schematic to start the program. If an simulation result file (.sdb) for the active simulation model exists, it is opened along with the DAY Post Processor. See also "Overview of File Formats" on page 424.

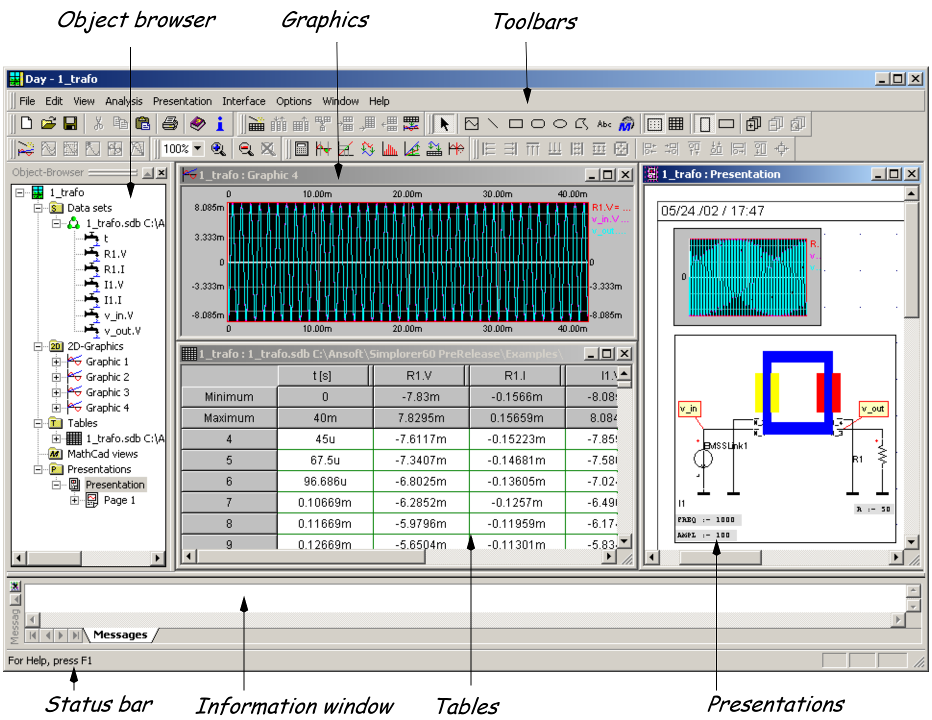
<u>This section contains information on</u>

- DAY Post Processor main window
- Overview of file formats
- Importing simulation data
- Creating and changing tables and graphics
- Analyzing of simulation data
- Creating presentations
- Exporting data

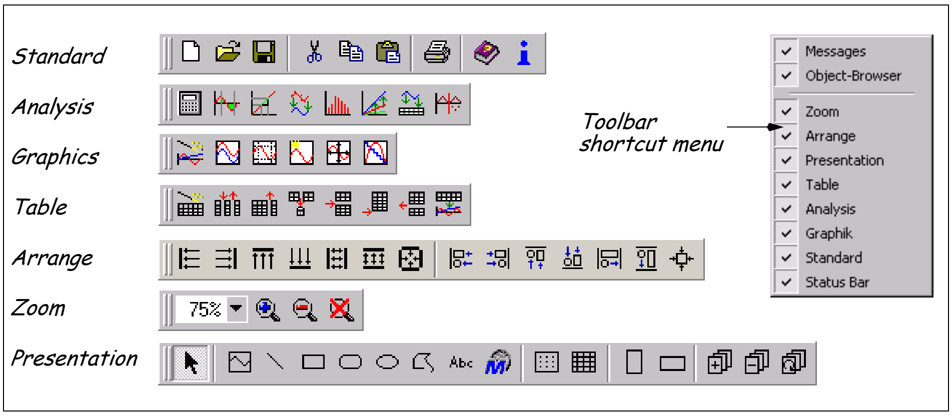
14.1.1 DAY Post Processor Main Window

The following window elements can be switched on and off with the commands on the OPTIONS menu:

Graphic, table, and presentation window	Presents the simulation data in various ways.
Object browser	Shows all displays (graphics, tables, presentations) and data sets in an analysis file.
Information window	Shows DAY Post Processor error messages while loading, running, and saving files.

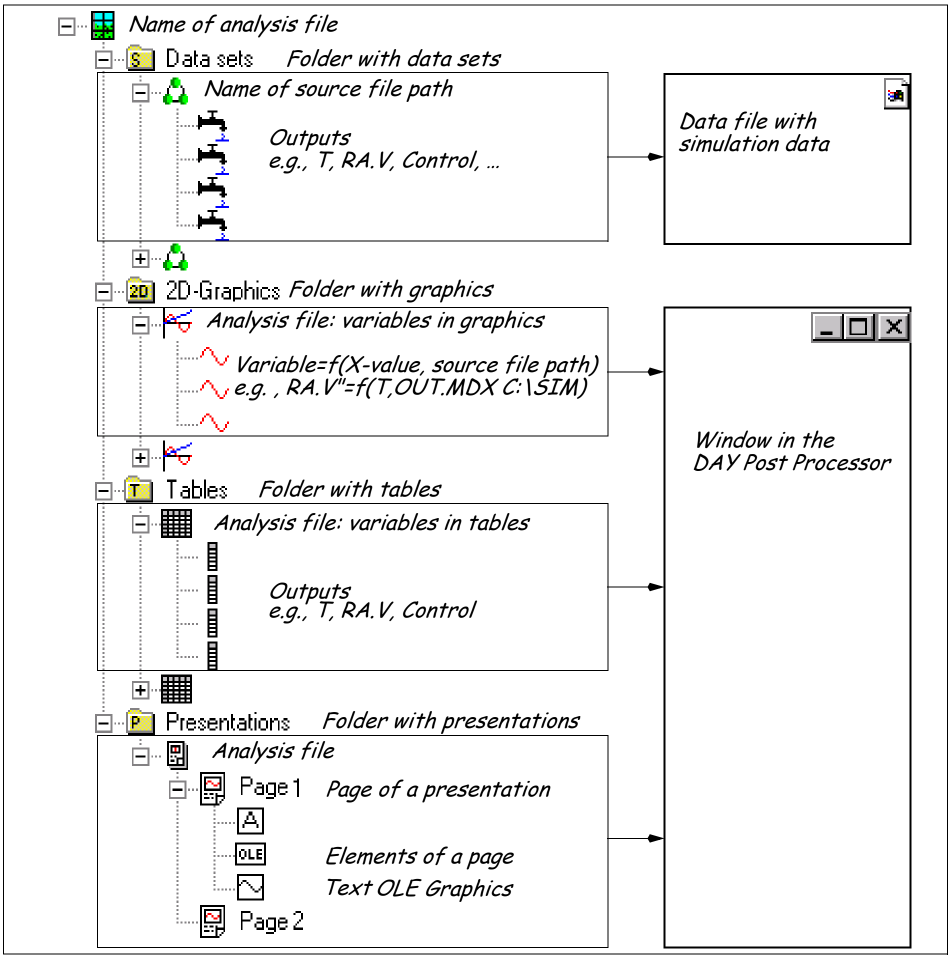


DAY Post Processor Toolbars

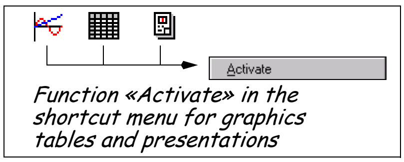


DAY Post Processor Object Browser

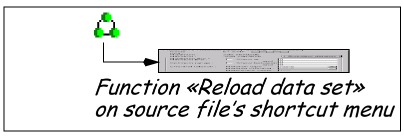
The Object browser displays all open files and their related entities, such as graphics, tables, and presentations, in a tree structure.



You can switch to the related window (graphic, table, or presentation) via **«Activate»** on the shortcut menu. You can also double-click an object name (Diagram, Table, Mathcad View) to activate the window.



Select **«Reload data set»** from the source file's shortcut menu to update simulation data in the displays.



DAY Post Processor Message Window

The Message window displays warnings and error messages from the DAY Post Processor, calculator, and interface. These messages, which help you find and resolve analysis errors, can be copied into the Clipboard and inserted into a separate editor.

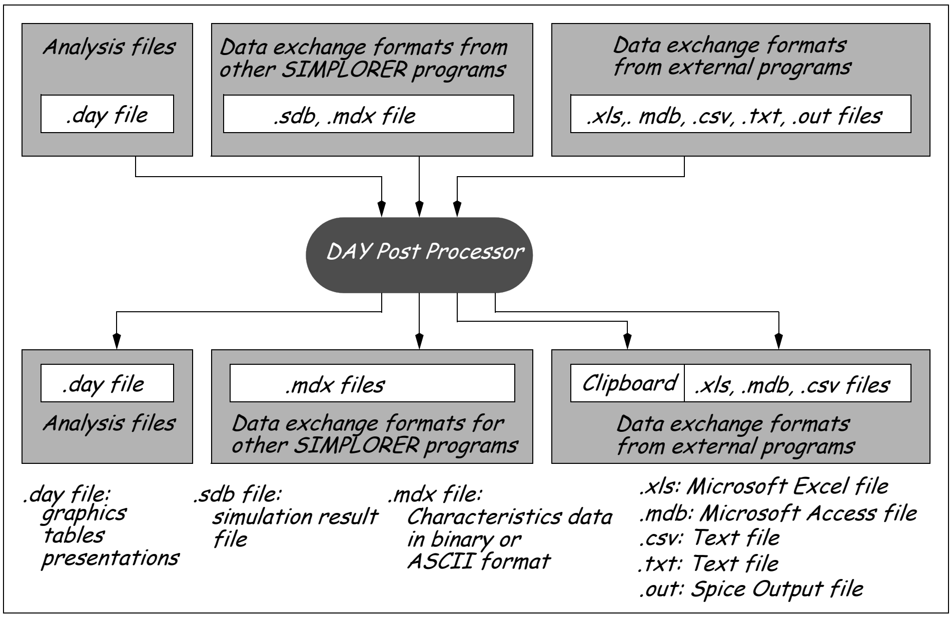
14.1.2 File Management

DAY Files contain all simulation data created in the DAY Post Processor, including summaries and illustrations. If you want to automatically update simulation data, these source files must be available when you relocate a .day file.

The SIMPLORER project management allows you to copy and move complete project data, provided that a project is used. See also 2.5 "Project Management with the SSC Commander" on page 14.

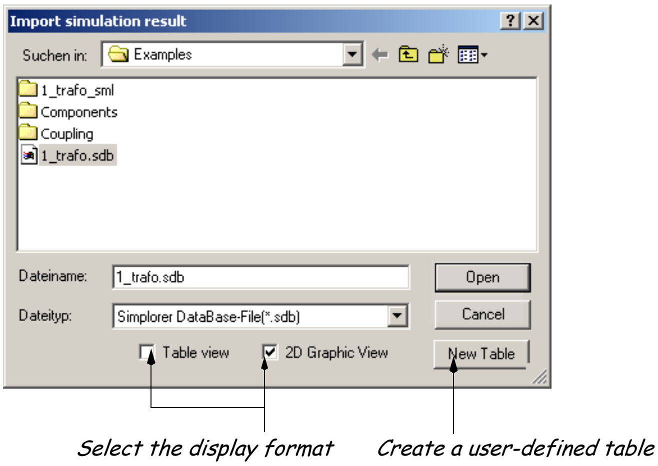
Overview of File Formats

Data files are created either during the simulation when you define outputs or after the simulation via the View Tool (FILE>SAVE SELECTIVE). The DAY Post Processor includes these data files in an analysis file, along with the graphics and presentations. You can export data files for SIMPLORER programs or other applications.



Creating a New Analysis

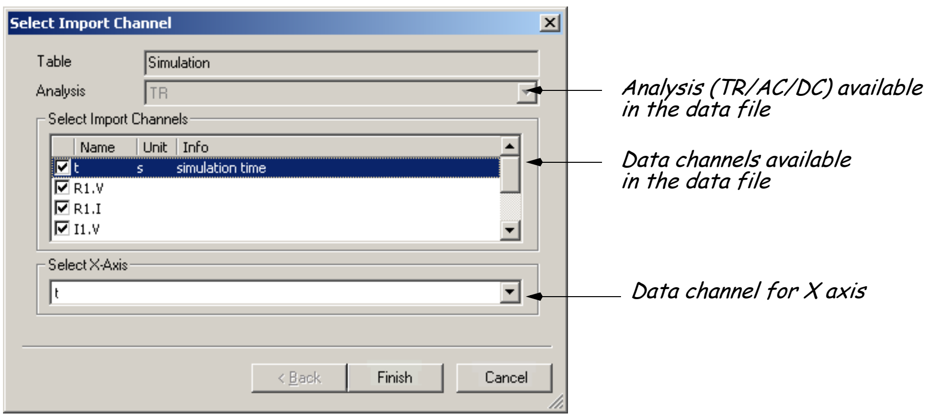
If you start the DAY Post Processor without selecting an existing analysis or data file, choose FILE>NEW to create a new analysis file. You can load any SIMPLORER data files (.sdb, .mdx, .xls, .mdb, .csv, .txt,.out) or create a table with new data. Define **«Table»** or **«2D Graphic»** or both as display format.



If you click <Cancel>, a new empty analysis file opens. The <New Table> command creates a new table with definable data channels and data sets. See also "Creating an Empty Table" on page 430.

Add Data to an Existing Analysis

To import additional simulation data in **one** analysis file, select the analysis file in the object tree, choose FILE>IMPORT SIMULATION RESULTS and select a data file (.sdb). Define **«Table»** or **«2D Graphic»** or both as display format.



The data (table and/or graphic) selected in the dialog box is added to the active analysis file.

A period (dot) must be used as decimal symbol for numbers.

If any other file is selected, the import function starts. This works especially well for non-SIMPLORER data formats. The DAY Post Processor can use any files in the appropriate form (i.e., with separators between the data fields and carriage returns between the data sets).

File Functions in the DAY Post Processor

File Menu

Command	Toolbar Symbol and Description
New	Creates a new data analysis file (DAY) or a new.
Open	Opens a data analysis file (DAY).
Close	Closes the active DAY file.
Save	Saves the data analysis file (DAY).
Save as	Saves the data analysis file under a new name.
Import simulation result	Opens an .sdb, .mdx, .mdk, .xls, .mdb, .csv, .txt, or out data file and add the data to the active analysis.
Export simulation result	Exports a .mdx, .xls, .mdb, or .csv data file. See also 14.1.12 “Exporting Data” on page 462.
Project	Switches to the SSC Commander.
Print	Prints the active file (graphic, table, or presentation).
Print preview	Shows the print preview of the active window (graphic, table, or presentation).
Page setup	Printer setup of the active printer driver.
«List of last opened files»	Shows the last opened data analysis files.
Exit	Closes the DAY Post Processor.

Displaying Properties of Simulation Data

You can display simulation data properties (in the source file) with the Edit>Properties simulation results command. The **«Automatic takeover of changed source files»** option updates the simulation data immediately when you load the file, and the **«Automatic export while saving document»** option updates the modified data in the source file.

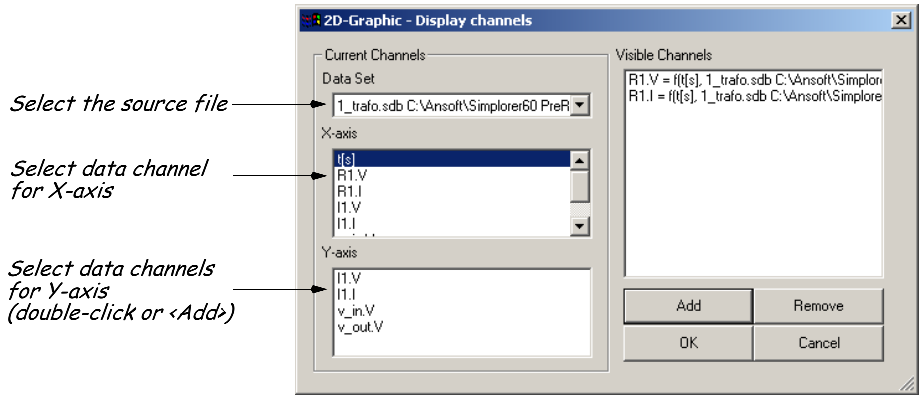
14.1.3 Creating Graphical Representations

There are two ways to create graphics for tables containing simulation data:

- Using all data channels in a table.
  Use the «Display as 2D-Graphic» command on the table's shortcut menu in conjunction with the Object browser.
- Using selected data channels in a table.
  Use the VIEW>2D-GRAPHIC NEW command.

Creating a Graphical Representation with Defined Data Channels

In the 2D-GRAPHIC NEW dialog, select the data channels to assign to the X- and Y- axis in a coordinate system.



Data set
The drop-down list shows the loaded result files for the selected analysis. Click the desired data set to select it.

X-axis
Selects one data channel for the X-axis.

Y-axis
Selects multiple data channel for the Y-axis. To add a data channel to the graphic, select the data channels in the list and click <Add>, or double-click the array entry.

Visible Channels
Lists data channels for the Y-axis. To remove a data channel from the graphic, select the data channel in the list and click <Remove>, or double-click the entry.

<Remove>
Removes the display for marked channels.

<OK>
Saves the active settings.

<Cancel>
Closes the dialog box without saving the changes.

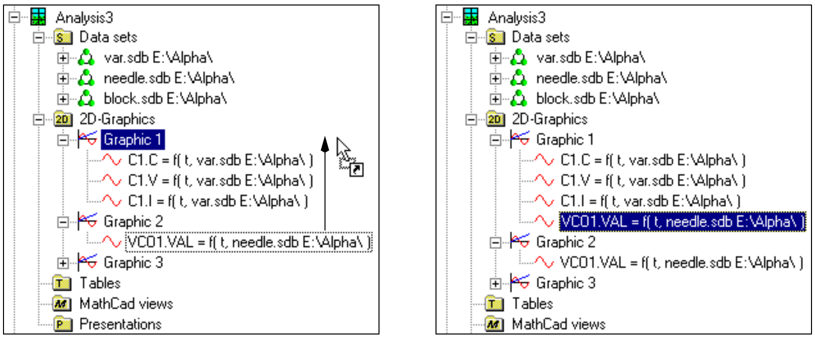
Insert Data Channels from Different Source Files into one Graphic

1 To import simulation results into **one** analysis file, choose FILE>IMPORT SIMULATION RESULT, or double-click a simulation data file in the project list of the SSC Commander.

If you inserted the data as a table in the analysis file, you can create a graphic from the table later.

3 Select the graphic, from which data channels should be copied, in the Object browser.

3 Drag the data channel you want from the selected graphic to the other graphic. The mouse cursor changes into a shortcut symbol when you drag over a valid insertion area.



Select the data channel and drag it to the other graphic

The additional data channel is inserted into the other graphic

4 Release the mouse button. The data channel is inserted in the coordinate system. The linked channels may have different dimensions, and you can adjust the size of the channels using the **«Scale»** command on the properties menu of the data channel.

Graphic and Zoom Toolbar Functions

You use the graphic toolbar to edit 2D graphics. It can be displayed using VIEW>TOOLBARS>GRAPHIC.

Symbol Function	Toolbar Symbol and Description	
New Graphic		Starts the VIEW>2D GRAPHIC NEW dialog. It creates a new graphic with data channels selected in the dialog.
Modify Channel		Opens a list with all data channels in the graphic. Click an entry to open the data channel property menu.
Add and remove channels		Opens the dialog to change the display of data channels. The command is equivalent to the graphic shortcut command **«Show/Hide Channels»**.
Modify coordinate system		Opens the dialog to modify the graphic coordinate system.

Symbol Function	Toolbar Symbol and Description	
Best representation		Sizes all data channels to place all signal amplitudes within one scope.
Show Cursors		Cursor tracing of the selected wave form; numerical values appear in the status bar (bottom right).

Zoom In and Zoom Out

You can zoom in (enlarge) selected areas in the graphical display. Hold the left mouse button, and drag the mouse over the area to be enlarged. You see a rectangle on the screen as you drag the mouse. When you release the mouse button, the area inside the rectangle is enlarged in relation to the active window size. You can incrementally zoom in or out, or restore the initial view, using the Zoom toolbar.

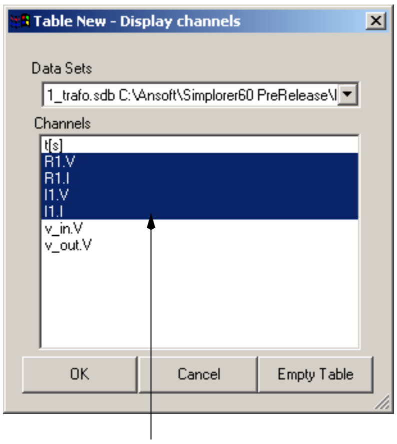
To fit all data channels to the size of the diagram, use the **«Best Representation»** command on the graphic's shortcut menu. The **«Real Size»** command removes all zoom settings and restores the normal display.

Symbol Function	Toolbar Symbol and Description	
Percent of scaling	75%	Defines the scaling size in presentation mode.
Zoom in		Enlarges the display in 2D graphics and presentations by one step.
Zoom out		Reduces the display in 2D graphics and presentations by one step.
Real size		Resets the zoom settings in 2D graphics and presentations.

14.1.4 Creating Tables

You can create tables of simulation data using the VIEW>TABLE NEW command or the **«Table new»** option on the tables shortcut menu in the Object browser.

Creating a Table with Simulation Data



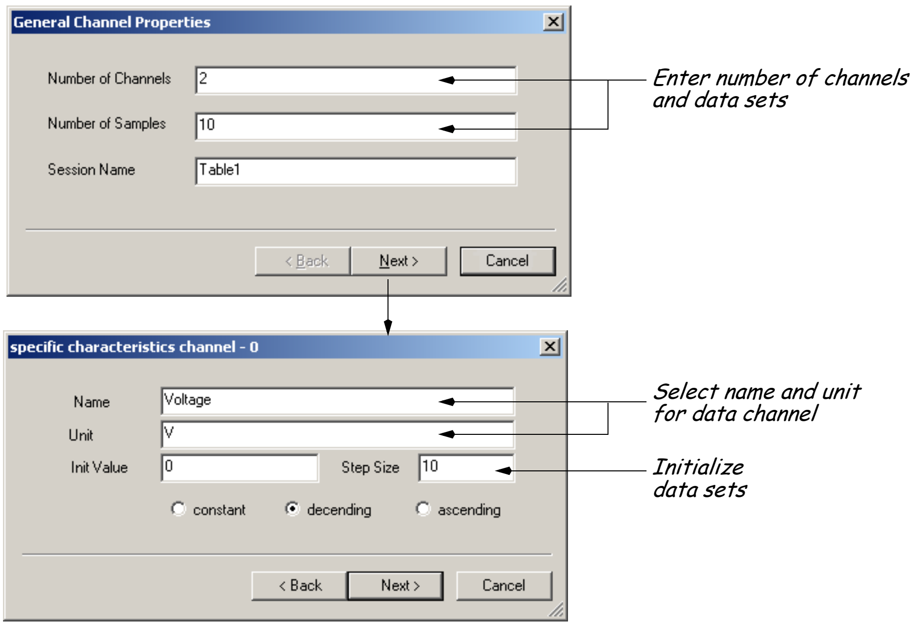
Select the data channels for the table

With the VIEW>TABLE NEW command, all data channels can be selected in the table dialog and combined into a table.

After selecting a source file from the drop down menu, you have to select the desired data channels in the «Channels» field. Display a table of the selected data channels in a new window by clicking on <OK>.

Creating an Empty Table

A new table with a predefined number of data channels and data sets can be created with VIEW>TABLE NEW <Empty Table>.



View Menu

Command	Toolbar Symbol and Description	
2D-GRAPHIC NEW		Creates a graphic from available simulation data sets. The data channel can be chosen from the list.
TABLE NEW		Creates a table from available simulation data sets or creates an empty table.
PRESENTATION NEW		Creates a new presentation.
MATHCAD NEW		Creates a new Mathcad element in a presentation.
ZOOM IN		Enlarges the display in 2D graphics and presentations by one step.
ZOOM OUT		Reduces the display in 2D graphics and presentations by one step.
REAL SIZE		Resets the zoom settings in 2D graphics and presentations.
GRID POINTS		Shows grid points in a presentation.
GRID LINES		Shows grid lines in a presentation.
PORTRAIT FORMAT		Sets portrait format for a presentation.
LANDSCAPE FORMAT		Sets landscape format for a presentation.
SHOW/HIDE CHANNELS		Inserts or deletes channels in a graphic or table.
PROPERTIES		Opens the properties dialog of the selected elements. Sets the properties for active graphic or table. See also 14.1.6 “Options and Settings for Graphics and Tables” on page 435.
OBJECT BROWSER	✓	Shows or hides the Object browser.
OUTPUT MESSAGES	✓	Shows or hides the information window.
TOOLBARS	✓	Shows or hides the Schematic toolbars: ZOOM, ARRANGE, PRESENTATION, TABLE, ANALYSIS, STANDARD, STATUS BAR.
STATUS BAR	✓	Shows or hides the status bar with help messages and key status display.

The grid lines, grid points, portrait format and landscape format options are available only in the presentation mode. See also 14.1.10 “Using Presentations” on page 454.

14.1.5 Managing Simulation Data

There are many ways to edit graphics and tables and to adapt the presentation to your needs. The following elements are described in the following paragraphs:

- Edit the data channel
- Edit the coordinate system
- Change the properties (colors and elements) of a graphic
- Edit Tables

Adding, Removing, and Deleting Data Channels

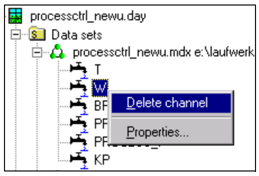
You can remove (hide), add (show), or delete data channels in analysis file presentations.

- Remove (Hide) and Add (Show)

The **«Remove»** option removes the data from all representations (graphic, table, presentation). It is possible to add them again with **«Add»**. You can find **«Remove»** and **«Add»** options in the shortcut menus of the data channels (Object browser or window) and in the SHOW/HIDE CHANNELS menu.

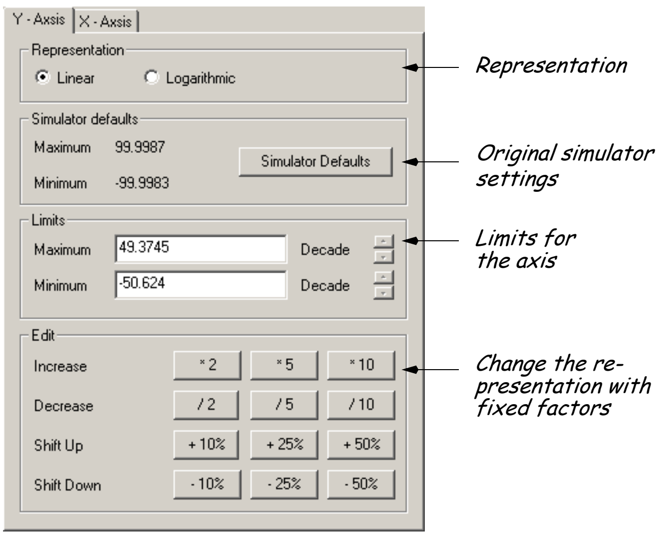
- Delete

The **«Delete»** option deletes the data from the source file. This means the data are no longer available for presentations.You can find **«Delete»** in the shortcut menu of the data channel in the source file.



It is not possible to update the simulation data if you delete a data channel because data cannot be updated after the number of data channels has changed.

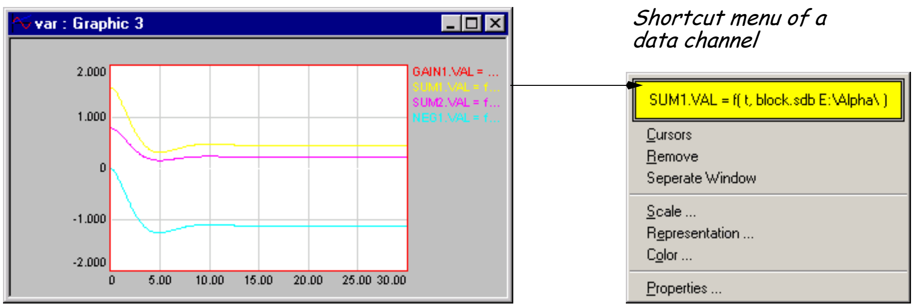
Editing the Coordinate System



You can set the size of the coordinate system and the representation (linear or logarithmic) independently for the X and Y axes.

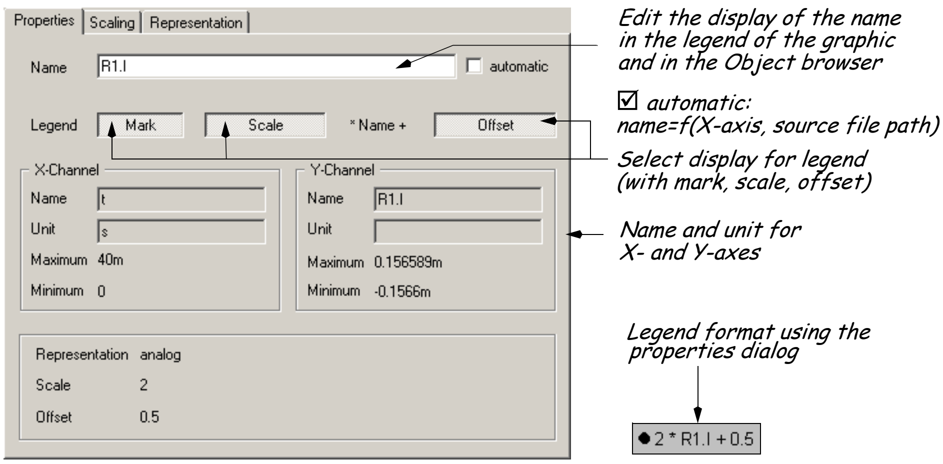
Editing Data Channels

You can find the commands to edit the properties of the data channel in the shortcut menu of the data channel. To open the shortcut menu click the data channel with the right mouse button in the graphic itself or click the related symbol in the Object browser.



<u>Properties Menu</u>

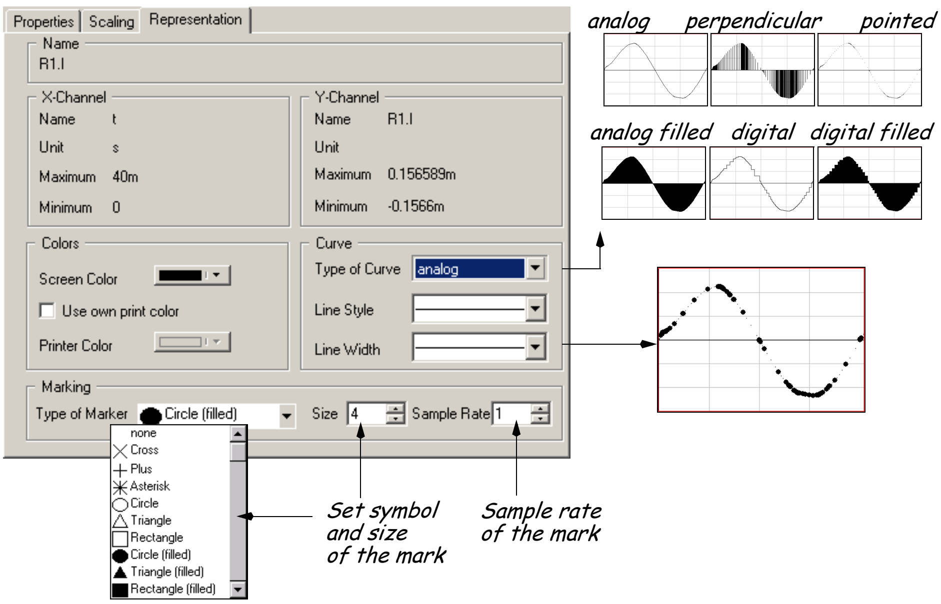
Change the names of a data channel



Change scale and offset

For a selected data channel, use the options in the «Scale» dialog box to set the scale factor and the offset for the Y-axis.

Edit curve properties



You can also change the line width and shape for the analog representation. Click a color box to open the color settings dialog. If you want to use your own print colors, be sure to **clear** the option **«Use screen colors»** in OPTIONS>2D SETTINGS **«Print colors»** and then to define a new color for each data channel.

Editing Tables

In addition to the View menu options, the TABLE toolbar also provides options to edit and modify data channels. If a row or column is selected, the options take effect for the entire row or column. You can change the data in a table by editing the content of a cell manually. Clicking in the table's cell starts the edit mode.

Symbol Function	Toolbar Symbol and Description
Table new	Creates a table from available simulation data sets or creates an empty table of the active graphic.
Display or hide Channel	Opens the dialog to display or hide channels of the active table.
Hide channels	Hides the selected channel, which can be displayed again.
Delete channel	Deletes the selected channel which cannot be displayed again.
Insert line	Inserts an empty line above the selected line.
Add line	Inserts an empty line at the end of the table.

Symbol Function	Toolbar Symbol and Description	
Delete line		Deletes the selected line which cannot be displayed again.
Create new graphic with all channels		Creates a graphic with all channels in the table.

14.1.6 Options and Settings for Graphics and Tables

To set options in the analysis file (.day) for screen display and print layout for diagrams, select OPTIONS>2D SETTINGS; for tables, select OPTIONS>TABLE SETTINGS.

The VIEW>PROPERTIES menu assigns these settings for the **active** window only. The contents of the individual menus depends on whether you select graphic or table representation.

To change names of diagrams and tables, click the name in the Object browser, and enter the new name when the prompt appears. Click outside of the name to close the edit mode.

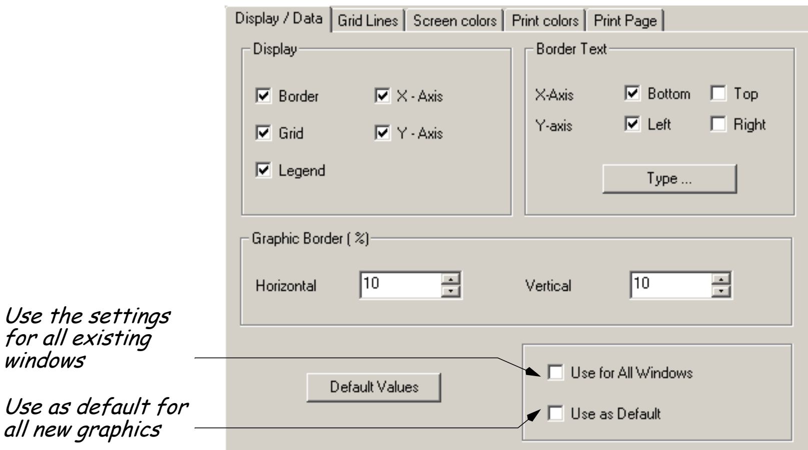
Option Menu▾

Command	Description
2D SETTINGS	Sets the properties for **all** graphics.
TABLE SETTINGS	Sets the properties for **all** tables.

For Graphics

Options 2D Settings or View Properties

Dialog Tab	Description
«Display/Data»	Changes display settings (frame, axis, legend).



Dialog Tab	Description
«Grid Lines»	Sets number and orientation for the grid lines of linear and logarithmic coordinate systems.
«Screen colors»	Sets display color for data channels and diagram itself. You can define a specific color for each data channel. Own color settings can be saved in a color scheme.
«Print colors»	Sets print colors. See also the SCREEN COLORS command.
«Print page»	Sets header, footer, and borders for printing.

For Tables

Options Table settings or View Properties

Menu	Description
«Channel»	Defines the number of digits displayed in the channels and the channel name. This menu is available only if you select a channel and choose VIEW>PROPERTIES.
«Table»	Sets grid color, width, display of minimum and maximum values of a data channel.
«Print page»	Sets the page layout, header, footer, and page margins.

14.1.7 Data Analysis and Processing

The DAY Post Processor comes with powerful tools to calculate new data channels. These tools include integration, differentiation, calculation of power and characteristics and the Fast Fourier Transformation (FFT) of data channels. The Data analysis commands are available in the ANALYSIS menu; most can also be selected from symbols on the ANALYSIS toolbar.

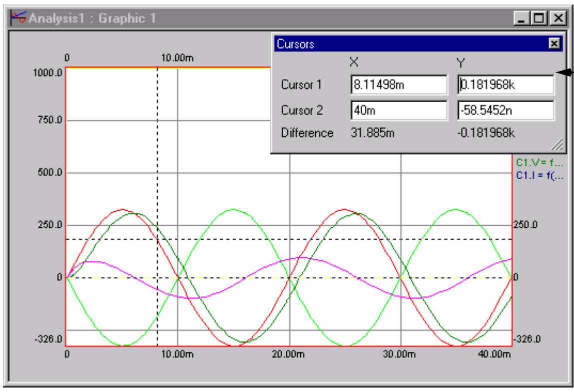
Analysis Menu ▾

Command	Toolbar Symbol and Description
CURSORS	 Cursor tracing of the selected wave form; numerical values appear in the status bar (bottom right).
CALCULATOR	 Performs arithmetic and logical operations on data from one or several channels in all analysis files.
INTEGRATION	 Integrates a selected data channel over another one.
DIFFERENTIATION	 Differentiates a selected data channel over another one.
SMOOTH	 Computes the mean value to smooth simulation or measurement data.

Command	Toolbar Symbol and Description
EQUIDISTANT	Adapts the data sets in a simulator file to a homogeneous time step.
FFT	Computes the frequency spectrum of a channel.
POWER	Calculate the power and Total Harmonic Distortion, a measure of the harmonic content, for any simulation quantity.
FLIP	Doubles an data channel axissymetrical around the Y axis for different X values.
CHARACTERISTICS	Compute parameters important in time domain system analysis.
MATLAB TOOLBOX	Opens the Matlab Toolbox dialog to exchange data between the DAY Post Processor and the Matlab workspace. See also 14.1.8 “Matlab Tool Interface” on page 449.

Cursors

If a curve has been drawn in a coordinate system, then the ANALYSIS>CURSOR command allows the user to fade in two cursors which can be moved by mouse or keyboard buttons. The cursor positions (x, y) are displayed in the cursor output window. The ANALYSIS>CURSOR command is also available on the shortcut menu of a data channel.



Cursor display in the graphical presentation

The TAB key switches between the cursors. If the mouse is used, the left mouse button activates cursor #1 and the right one activates cursor #2. The cursors move from one data set entry to the next one in sequence. If the data set is very large and the keyboard buttons are used, press the SHIFT and the CTRL key to increase the cursor speed. If several curves are displayed in one window, press the UP or DOWN ARROW key to switch between the curves.

If you know one the coordinates of a specific value, you can enter the value directly in the cursor output window. After TAB the cursor is placed at the entered position automatically.

Calculator

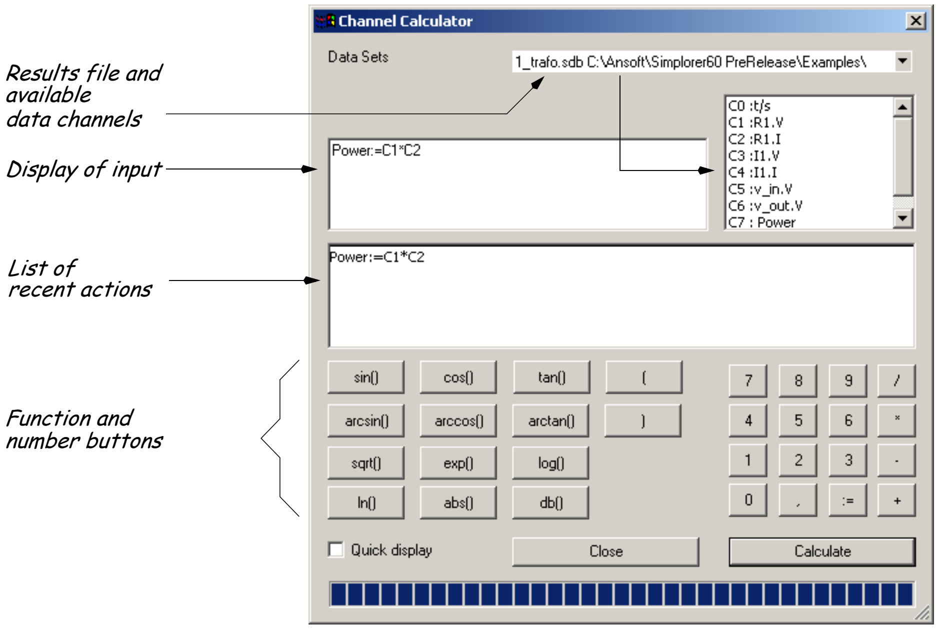
With the calculator users can perform arithmetic and logical operations on data from one or several channels in all analysis files. The resulting data can then be assigned to a new channel. To assign an expression to a channel, always use the assignment operator (:=). This operator and other commonly used operators are already available in the pocket computer keypad integrated in the channel calculator dialog. Existing channels can be selected with a double-click.

Operator		**Mathematical function**	
:=	Assignment	SIN ARC-SIN SINH	Sine, Arc sine, Sine hyperbola
( )	Parenthesis	COS ARC-COS COSH	Cosine, Arc cosine, Cosine hyperbola
*	Multiplication	TAN ARTAN TANH	Tangent, Arc tangent, Tangent hyperbola
/	Division	SQRT	Square root function
+	Addition	SQU	Square function
–	Subtraction	EXP	Exponential function
<	lower than	ABS	Absolute value function
>	greater than	LN	Natural logarithm function
< > or ><	Not equal	LOG	Decadic logarithm function
<= or =<	lower than or equal	RAD	Conversion degrees to radians
>= or =>	greater than or equal	GRD	Conversion radians to degrees
=	equal	GEL50	Conversion degrees el. to seconds at 50 Hz
AND	Negation	GEL16	Conversion degrees el. to seconds at 16 2/3 Hz
OR	Conjunction	GEL	Conversion degrees el. to seconds at 1 Hz
NOT	Disjunction		

Operating Elements of Channel Calculator

Data Sets

Select a data file from the drop down menu. The right frame displays the existing data channels with their names, channel numbers, and units.



Input field

Accepts input of a mathematical expression. A channel calculation formula may contain up to 255 characters. Click a function or number key to transmit the function or value into the edit line. Double-click a data channel name to transmit the corresponding string into the input field. Click once inside the input window to enter input directly with the keyboard.

Sample input: C1:=C1*10+20; C2:=C1+C2

Quick display

☑ Computation results are immediately integrated into the corresponding presentation (table).

<Calculate>

Shows the results of a computation. If the values of the computation must be stored in a separate channel, the target channel must be specified. The number of the target channel must be greater than the number of the currently used channel. In this case the results of the channel operation are stored in the defined new channel.

<Close>

Closes the channel calculator.

Calculator Error Messages

Calculator error messages are displayed in a brief form “Error:xx” in the input window. The following list shows the error messages and possible causes of the errors:

Error	Description
1	mathematical error, division by zero
2	mathematical error, illegal argument SQRT()
3	mathematical error, illegal argument LN()
4	mathematical error, illegal argument LOG()
5	mathematical error, illegal argument ARCCOS()
6	mathematical error, illegal argument ARCSIN()
7	lexical error, character out of lexical range
8	syntax error, the expression contains unrecognized syntax
9	error in writing temporary file
10	error in reading temporary file
11	internal error support request necessary!
12	mathematical error, illegal argument SQU()
13	no memory for internal functions
14	expression too large
15	variable list overflow!
17	mathematical error, illegal argument ARCTAN()
18	mathematical error, illegal argument EXP()
19	mathematical error, illegal argument SINH()
20	mathematical error, illegal argument COSH()
21	mathematical error, illegal argument TANH()
22	mathematical error, illegal argument FMOD()
96	variable error! You tried to combine a channel with an unknown or undefined variable.
97	Channel could not be generated! All 50 channels are used or this operation exceeds system memory.
98	Channel allocation error. You tried to use a nonexisting channel for computation.
99	index error, You tried to use the nominator "C" without an appropriate channel number index. This error also occurs at an attempt to assign a channel to a single variable - e.g.: Reg:=C1+10

Differentiation and Integration

Carries out a differentiation/integration of a data channel over a specified in a specified interval.

Data sets
Select the data file.

Differentiation/Integration of
Select the data channel (quantity) to be analyzed.

Over
Select the reference data channel (quantity).

Interval
Specify the interval the operation is applied to.

Cursor position
☑ Interval limits are set by the cursor (if cursor mode is active).

Complete area
☑ The whole curve is used.

Values
☑ Calculation for an individual interval according to a start and a stop value.

Destination
Select the destination device.

Overwrite channel
☑ The selected channel is overwritten by the new values.

Create new channel
☑ Specify name and unit of the resulting channel.

<OK>
Start the computation.

<Cancel>
Closes the input dialog without saving.

Quick display
If the box is checked, the computation results are immediately integrated into the corresponding presentation (table).

Smooth

Mean value computation and channel smoothing. This function is used to smooth simulation or measurement data.

Data Sets
Select the requested data set.

Source
Select the channel to be smoothed.

Filter size
Number of samples.

Destination channel
☑ Overwrite channel: The original channel is overwritten with the smoothed data set.

☑ Create new channel: The smoothed data set is stored in a new channel.

Quick display
If the box is checked, the computation results are immediately integrated into the corresponding presentation (table).

Equidistance Function

The Equidistance function adapts the data sets in a simulator file to a homogeneous time step, allowing you to easily compare several simulation runs, since a data set is always available for the same time step. The following options are available to adapt the time step:

Data Sets
Select the requested data set.

Sampling Rate
☑ Sampling rate: Adapt data channels to the given time step (sampling rate).

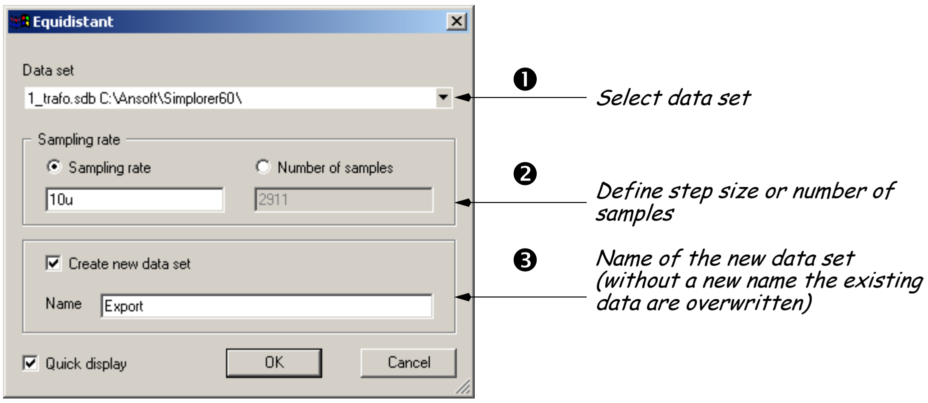
☑ Number of samples: Adapt data channels to the given number of samples (reduce or enlarge).

You can also create a new data set. If you do not create a new data set, SIMPLORER automatically overwrites existing data with the equidistant values.

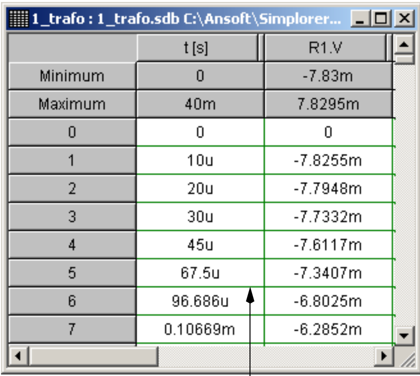
Quick display
If the box is checked, the computation results are immediately integrated into the corresponding presentation (table).

<OK>
Start the computation.

<Cancel>
Close the input dialog without saving.

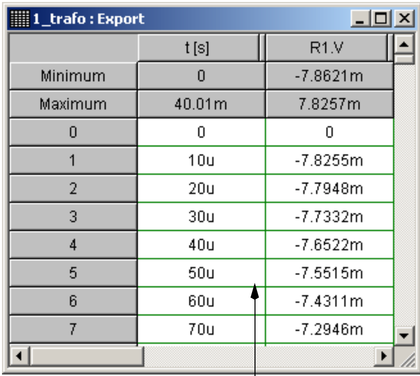


After the transformation (with sampling rate or number of samples), all data channels included have the same fixed time step.



	t [s]	R1.V
Minimum	0	-7.83m
Maximum	40m	7.8295m
0	0	0
1	10u	-7.8255m
2	20u	-7.7948m
3	30u	-7.7332m
4	45u	-7.6117m
5	67.5u	-7.3407m
6	96.686u	-6.8025m
7	0.10669m	-6.2852m

Data before calculating



	t [s]	R1.V
Minimum	0	-7.8621m
Maximum	40.01m	7.8257m
0	0	0
1	10u	-7.8255m
2	20u	-7.7948m
3	30u	-7.7332m
4	40u	-7.6522m
5	50u	-7.5515m
6	60u	-7.4311m
7	70u	-7.2946m

Data after calculating

FFT (Fast Fourier Transformation)

The Fast Fourier Transformation algorithm (Cooley-Tukey algorithm) computes the frequency spectrum of a channel. See also: Cooley J.M., Tukey J.W.: An Algorithm for the Machine Calculation of Complex Fourier Series. Math. Comp. 19, 297-301, 1965.

The resulting transformation is displayed in a separate window as a table or graph, depending on your settings.

The FFT computation algorithm requires the time domain data to be equidistant (equally spaced), so that there is an equal time interval between each data set entry. SIMPLORER simulation files are not necessarily equidistant due to the variable simulation time step. Therefore, the FFT algorithm automatically generates equidistant values while simultaneously adapting the number of entries in the time window to an integer power of 2.

Data Sets
Select the requested data set.

Source
Select the channel to be analyzed.

Over
Select the reference channel.

Time Window
If the **«Automatic»** box is checked, the whole time or curser range is used, otherwise you can define an interval.

Filter
Amplitude values smaller than x% of the maximum are not displayed.

% Evaluation
Maximum value of amplitude is 100%, otherwise define a new value as maximum.

Result
Select the format of the result, absolute value and phase, or real and imaginary part.

Window

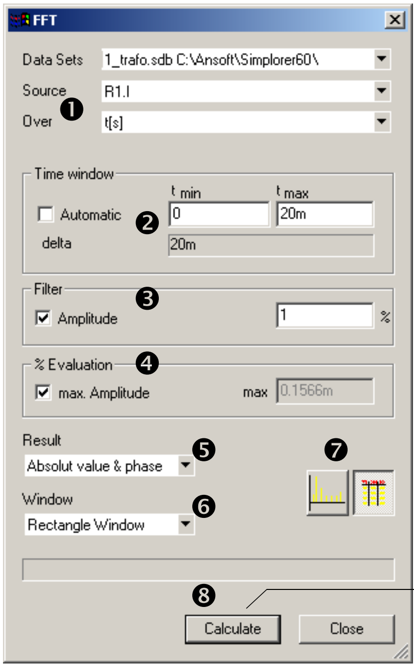
Select the window used for the FFT calculation.

<Calculate>

Start the computation.

<Close>

Close the FFT dialog without saving.



❶ *Select simulation data*

❷ *Set Time window*

❸ *Set amplitude filter*

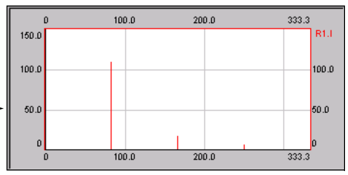
❹ *Set% for maximum amplitude*

❺ *Select the format of the result (absolute value & phase or real & imaginary part)*

❻ *Select the format of the result (absolute value & phase or real & imaginary part)*

❼ *Select Graphical or tabular representation of FFT in the new window*

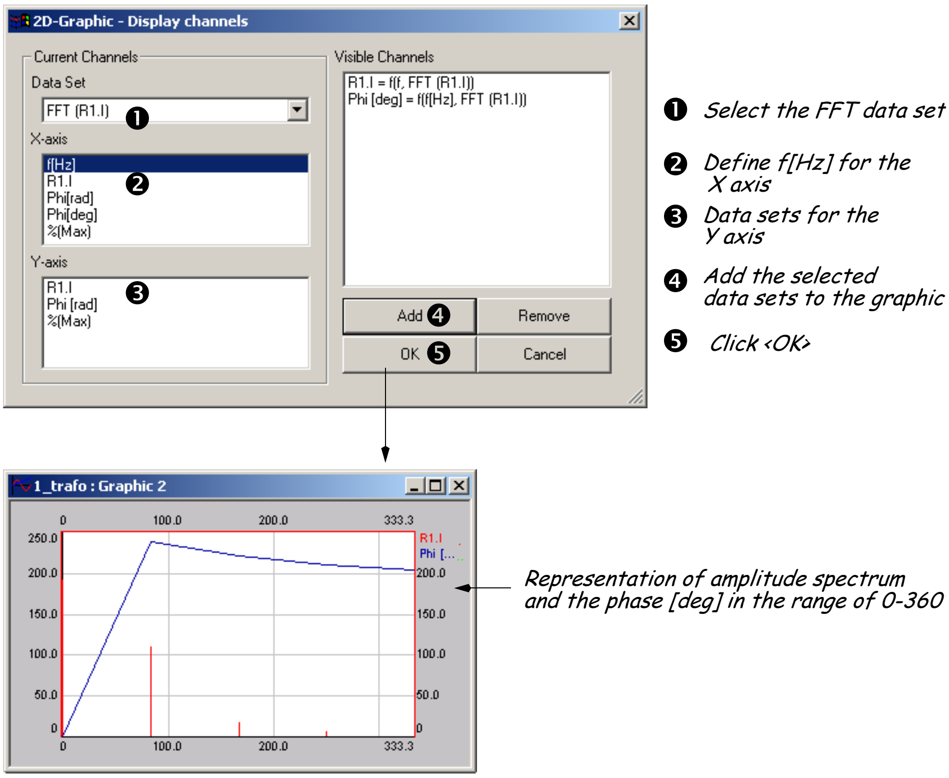
❽ *Start calculation*



After you click <Calculate>, only the amplitude spectrum is displayed. Phase (deg or rad) and proportional evaluation for amplitudes must be displayed separately.

When the computation is finished, close the FFT dialog. The results are displayed in the background. If you do not close the FFT dialog, you cannot perform any other function.

Choose VIEW>SHOW/HIDE CHANNELS to add other curves, such as the phase, to the 2D Graphic.



Power

Power and FFT are always analyzed for the time window defined by the cursors for a selected data set.

All harmonic waves are considered for the power computation.

Data Sets
Select the requested data set.

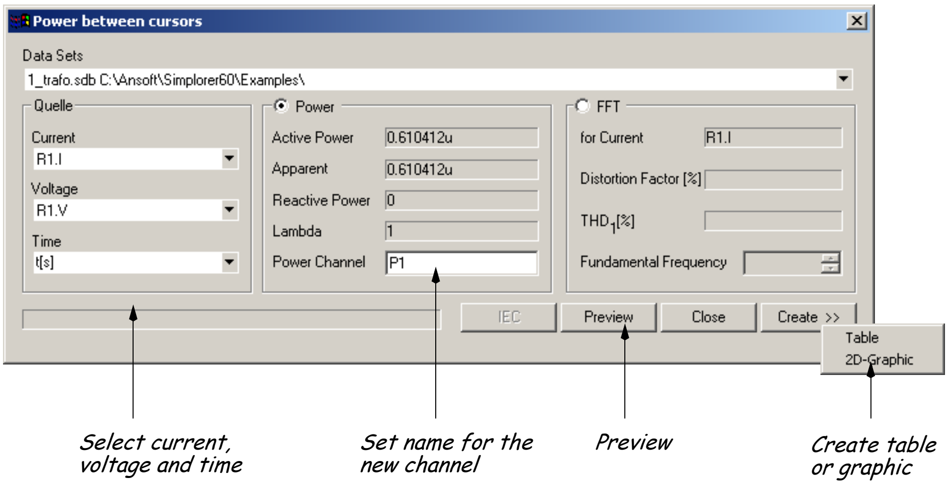
Source
Select the channels (current and voltage) to be analyzed.

Power
Select the function «Power». The active, apparent, and reactive power of the selected quantities are computed automatically.

<Preview>
Open a display within the dialog.

<Close>
Close the Power dialog box.

<Create>
Open the dialog to select table or graphic. The calculated values are displayed in a separate window in the selected presentation form.



Flip

The function mirrors values of a data channel at a specific axis and adds the data to the existing values.

Data Sets
Select the requested data set.

Flip Axis
Select the flip axis. You can choose between the Y axis (x=0), Simulation end time (x=Tend), or any other value (x=value).

Destination
Selects the kind of creating the results. You can choose between **«New Session»** (a new data set is created), or **«Overwrite old Channels»**. You can define a name for the new session.

Quick display
If the box is checked, the computation results are immediately integrated into the corresponding presentation (table).

<OK>
Start the computation.

<Cancel>
Close the input dialog without saving.

Calculating Total Harmonic Distortion (FFT Analysis)

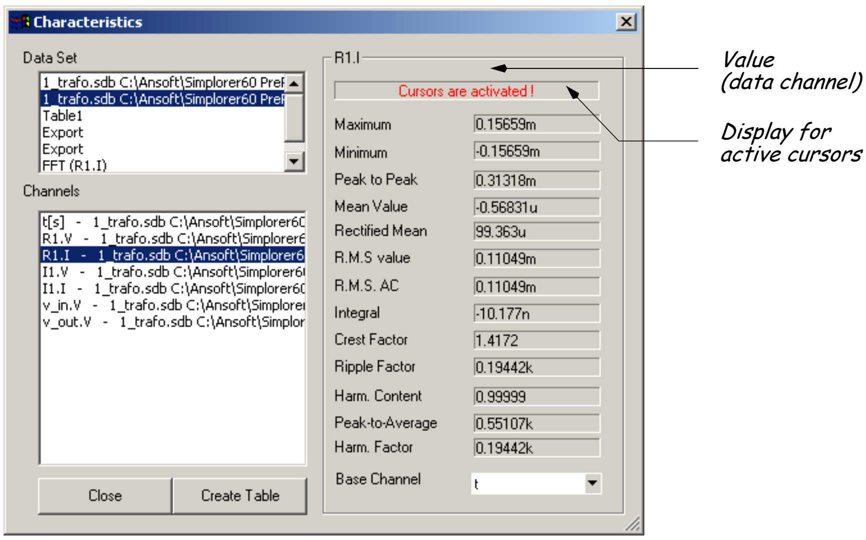
With the ANALYSIS>POWER menu in the DAY Post Processor, you can calculate the Total Harmonic Distortion, a measure of the harmonic content, for any simulation quantity. However, it is necessary to define a time period (through cursors) which is used for computation the Total Harmonic Distortion.

$$k = \frac{\sqrt{\sum_{n=2}^{\infty} A_n^2}}{\sqrt{\sum_{n=1}^{\infty} A_n^2}} \qquad THD_1 = \frac{\sqrt{\sum_{n=2}^{\infty} A_n^2}}{A_1}$$

The <Create> button opens the dialog to select table or graphic. The calculated values are displayed in a separate window in the selected presentation form.

Characteristics

The CHARACTERISTICS command includes tools to compute parameters important in time domain system analysis. Select first a data set and then a data channel by a mouse click in the window below. The characteristics, displayed in the table, relate always to the last selected data channel.



The <Table> command compiles the characteristic values of all selected data channels in a table. If cursors are active, then they represent the boundary values on the X axis for the parameter computation.

Computed Characteristics

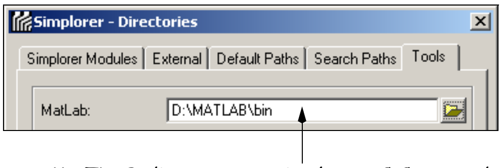
Meaning	Equations
Maximum value	$x_{max} = \max(x_1, x_2, \ldots, x_n)$
Minimum value	$x_{min} = \min(x_1, x_2, \ldots, x_n)$
Peak-to-peak value	$\Delta x_{pp} = \max(x_1, x_2, \ldots, x_n) - \min(x_1, x_2, \ldots, x_n)$
Mean value	$x_{mean} = \frac{1}{N} \cdot \sum_{i=1}^{N} x_i$
Mean absolute value	$\|x\|_{absmean} = \frac{1}{N} \cdot \sum_{i=1}^{N} \|x_i\|$
RMS value	$x_{RMS} = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (x_i^2 \cdot \Delta i)}$
AC RMS value	$x_{ACRMS} = \sqrt{(x_{RMS}^2 - x_{mean}^2)}$
Integral	$X = \sum_{i=1}^{N} x_i \cdot \Delta i$
Ripple	$x_{ripple} = \sqrt{\frac{x_{RMS}^2}{x_{mean}^2} - 1}$
Ripple factor	$k_{ripple} = \frac{x_{max} - x_{min}}{\|x_{mean}\|}$
Harmonic content	$h = \sqrt{1 - \frac{1}{x^2{}_{ripple}}}$
Form factor	$f = \frac{x_{RMS}}{x_{mean}}$
Crest factor	$c = \frac{\max(\|x_{max}\|, \|x_{min}\|)}{x_{RMS}}$

14.1.8 Matlab Tool Interface

The Matlab Tool interface allows direct data exchange between the DAY Post Processor and the Matlab Workspace and starts Matlab commands in the DAY Post Processor. Matlab provides a wide variety of mathematical functions, calculations with complex numbers and 3D graphics presentation.

Requirements

To use the interface, Matlab 5.3 or later must be installed on your PC. In addition, in the SSC Commander, you must enter the Matlab directory path to the *Matlab.exe* file in the OPTIONS>PROGRAM DIRECTORIES **«Tools»** (…\MATLAB\BIN) menu.



MATLAB directory set in the SSC Commander

Starting the Matlab Toolbox

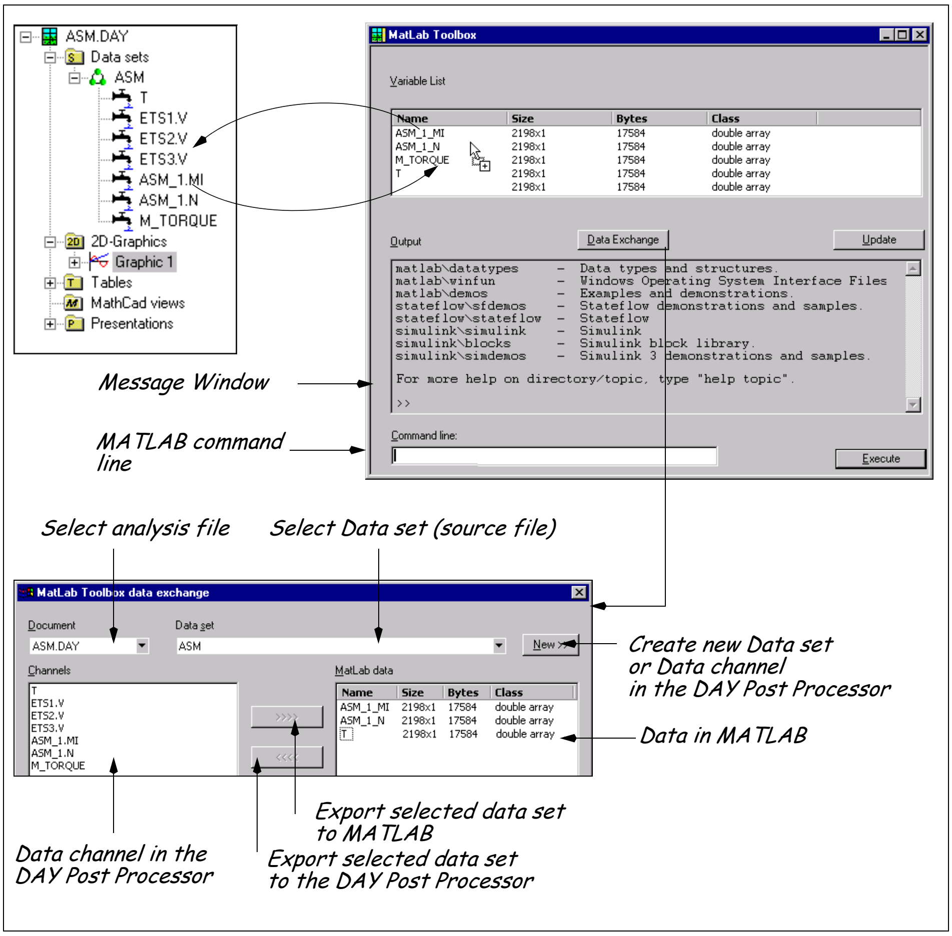
Start the Matlab toolbox in the DAY Post Processor with ANALYSIS>MATLAB TOOLBOX. The Matlab toolbox window opens.

Exchanging Data between the DAY Post Processor and Matlab

There two options for exchanging data between programs:

- Drag-and-Drop from the Object browser.
- The **«Data exchange»** function.

Select the data channel in the Object browser, and drag it into the Matlab toolbox window. (You can export data to the DAY Post Processor in the same way, by dragging a channel from the Matlab toolbox into the Object browser.) To exchange data through the dialog, select <Data exchange>.

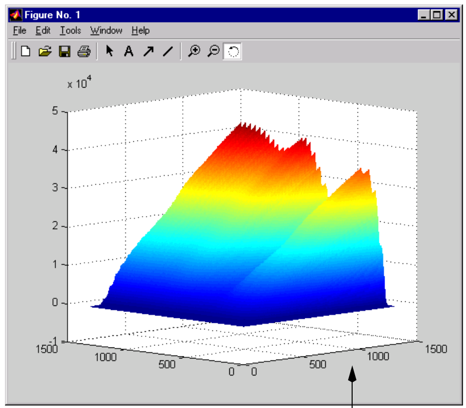


Modifying Data in Matlab

You can enter and execute all Matlab commands in the command line via <Enter> or <Execute> in the Matlab Toolbox window. The Matlab outputs are displayed in the message window. If possible, avoid Matlab commands that have outputs, and use a semicolon ";" at the end of a command. After each Matlab command, each data transfer, and each <Update>, variable list contents are updated.

You can also execute commands in the Matlab command window, which includes the variables and is open in the background.

No data modified in the Matlab workspace is saved by the DAY Post Processor. To save this data in an Analysis file, you must manually copy the data into the DAY Post Processor.



3D-representation of exported simulation data in Matlab

Data Types

The DAY Post Processor and Matlab represent data different from each other.DAY interprets all data sets as numbers with double precision without imaginary parts in (one dimension) column vectors. Matlab also interprets other data types with other precision and imaginary parts (complex numbers) in n-dimension data fields. The following conventions are valid for directly copying data between the DAY Post Processor and Matlab:

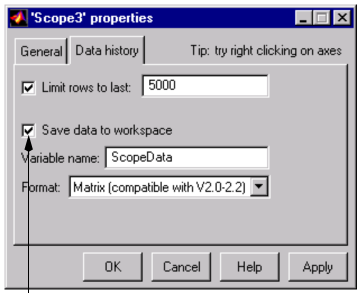
In MATLAB:	Changes to in the DAY Post Processor:
1 1 x 1 scalar, double, real	1 1 x 1 scalar, double, real
1 1 x 1 scalar, double, complex	2 1 x 1 scalars, double, real
1 n x 1 column vector, double, real	1 n x 1 column vector, double, real
1 n x 1 column vector, double, complex	2 n x 1 column vectors, double, real
1 1 x n row vector, double, real	n 1 x 1 scalars, double, real
1 1 x n row vector, double, complex	2n 1 x 1 scalars, double, real
1 m x n matrix, double, real	n m x 1 column vectors, double, real
1 m x n matrix, double, complex	2n m x 1 column vectors, double, real
1 m x n x o numeric array, any Type, …	not used

To copy other data types they must be converted into double exact matrices in Matlab. Each vector from the DAY Post Processor retains its properties in Matlab.

Simulating with the Simulink Interface and Data Evaluation with the Matlab Tool Interface

When you want to evaluate data created from a simulation coupling between SIMPLORER and Matlab/Simulink, follow these steps:

1. Start the DAY Post Processor.
2. Open or create an Analysis file.
3. Choose ANALYSIS>MATLAB TOOLBOX to start the Matlab Toolbox in the DAY Post Processor.
4. Start Simulink from the Matlab command window.
5. Edit the Matlab/Simulink Model. Define the Simulink Scopes in PROPERTIES>DATA HISTORY and the variables for outputs in the MATLAB Workspace.



Send simulation data to the workspace

When the simulation is complete, all data are available in both the Matlab Workspace and in the DAY Post Processor-Matlab Toolbox, and can be edited with Matlab commands or copied with drag-and-drop into the DAY Post Processor.

14.1.9 Mathcad Calculations

The SIMPLORER-Mathcad interface allows you to exchange data directly between the DAY Post Processor and Mathcad, and allows you to run Mathcad functions from within the DAY Post Processor.

Requirements

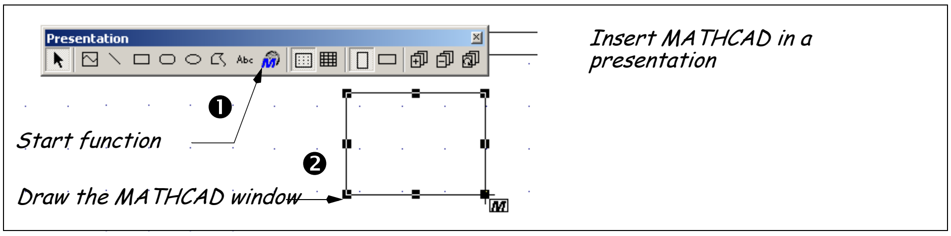
To use the interface, Mathcad Professional 7.0 or later must be installed or registered on your PC. With a proper installation of Mathcad, the VIEW>MATHCAD NEW command and the Mathcad symbol on the Presentation toolbar are available automatically.

Starting Mathcad

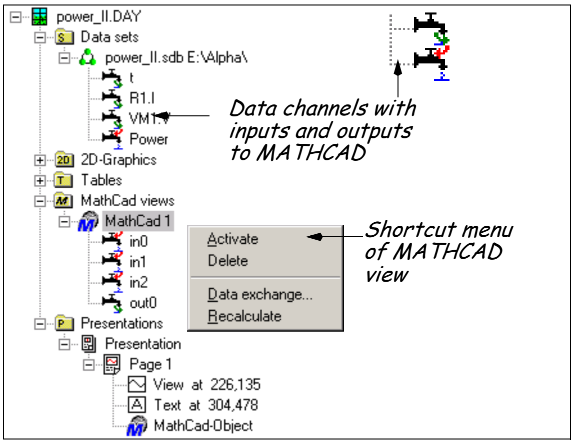
You have two options to integrate Mathcad in the DAY Post Processor: as a separate window or as an embedded object in a presentation.

The VIEW>MATHCAD NEW command starts Mathcad in a separate window.

Click the Mathcad symbol in the Presentation toolbar to insert a new Mathcad object in a presentation. You can also insert an existing Mathcad file with EDIT>OLE-OBJECTS>INSERT NEW OBJECT **«Create from File»**. The Mathcad window opens with its menus and toolbars.

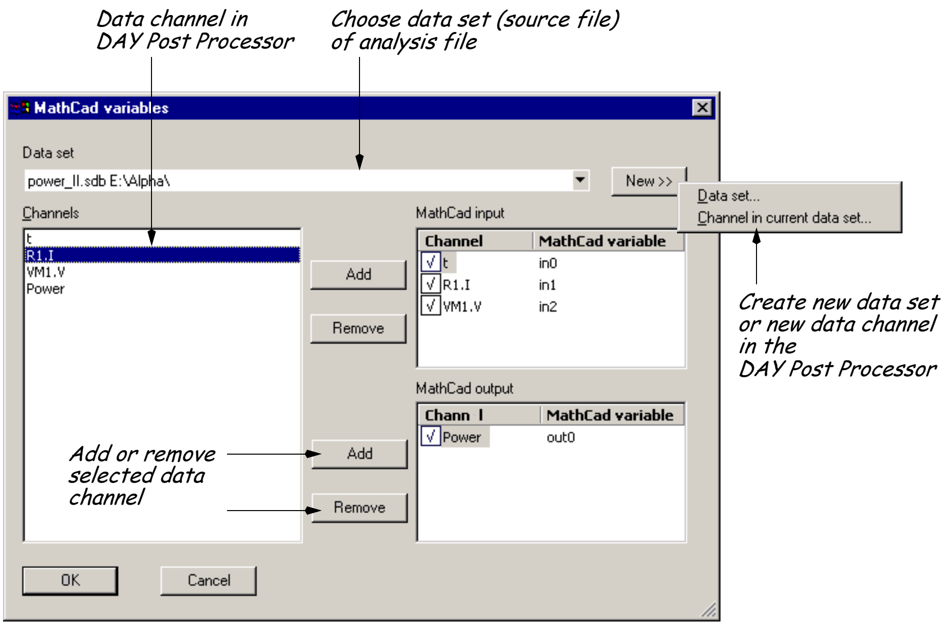


Exchanging Data between the DAY Post Processor and Mathcad

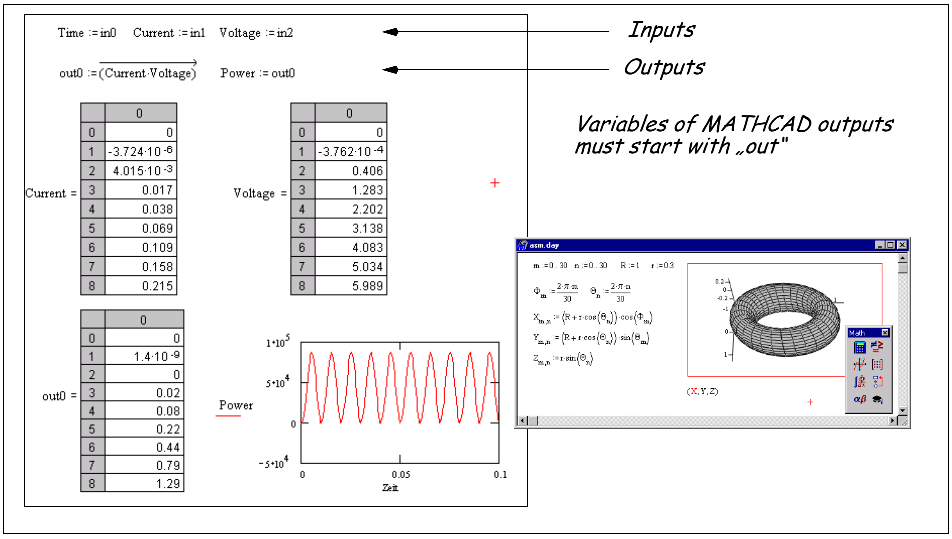


The Mathcad Dialog controls the data exchange between both programs. Choose **«Data exchange»** in the shortcut menu of the Object browser or MATHCAD>DATA EXCHANGE to open this dialog.

The **«Recalculate»** command updates equations and displays on the Mathcad Sheet; the **«Activate»** command puts the Mathcad window into the foreground.



Imported SIMPLORER data is displayed as a vector in Mathcad. Variables defined in the **«MathCad input»** window can be used for calculations in Mathcad. Any Mathcad function can be applied, provided that the proper data format is used. Only scalar vectors can be exported back into the DAY Post Processor; other data formats are not supported.



14.1.10 Using Presentations

In the presentation mode you can assemble tables, graphics, text, and drawing elements and print them together on one page. All changes in the source data are updated in the presentation automatically.

Creating a Presentation

In an analysis file only **one** presentation can be created, but the presentation can contain several pages. Use PRESENTATION>PRESENTATION NEW or **«Presentation new»** in the shortcut menu of the Object browser to create a new presentation. When you use VIEW>PRESENTATION NEW, if there is already a presentation for the analysis file, a new **view** is opened. You can insert different elements on each page of a presentation.

Inserting Elements in Presentations

You can insert elements either with the PRESENTATION>TOOLS commands or the symbols on the toolbar. The table below shows the symbol functions of the PRESENTATION toolbar:

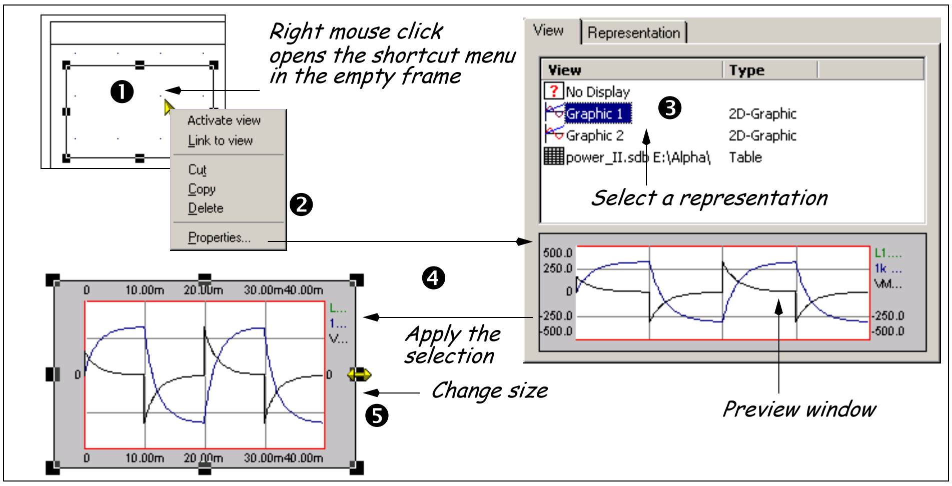
Function	Toolbar Symbol and Description
SELECT	Starts select mode (to alter or move shapes, lines, or text blocks).
DATA TOOL	Starts the data tool to define an area where a graphic or table can be placed.
LINE	Draws a line.
RECTANGLE	Draws a rectangle.
ROUNDED RECTANGLE	Draws a rectangle with rounded corners.
ELLIPSE	Draws an ellipse.
POLYGON	Draws a polygon.
TEXT	Starts the text mode.

A presentation can contain the following elements:

- Tables and graphics with simulation data
- OLE Elements (i.e., a circuit from Schematic)
- Text

Insert Simulation Data (Graphics and Tables)

1. Choose **«Activate»** in the shortcut menu of the Object browser to switch to the presentation window.
2. Start the Presentation tool with PRESENTATION>TOOLS VIEW or click the symbol in the Presentation toolbar.
3. The cursor changes into the symbol of the view tool. Holding the left mouse button, draw a rectangle on the presentation page and release the mouse button to create a new empty frame for a graphic or table.
4. Open the properties dialog with the shortcut menu of the empty frame; you see all graphics and tables of the analysis file collected in a list. Select the desired element in the list. The preview window shows a preview of the selected element. Click <OK> to copy the graphic or the table into the empty frame. You can change the size of the frame by pulling the markers with the mouse.



Insert Drawing Elements

1. Choose **«Activate»** in the shortcut menu of the Object browser to switch to the presentation window.
2. Start the draw mode with the PRESENTATION>TOOL command or click a symbol in the Presentation toolbar.
3. The cursor changes to a cross hair (or a pen for drawing polygons). Draw the element shape by clicking to set the corners. Finish the draw mode by clicking outside of the element.

14

Insert Text

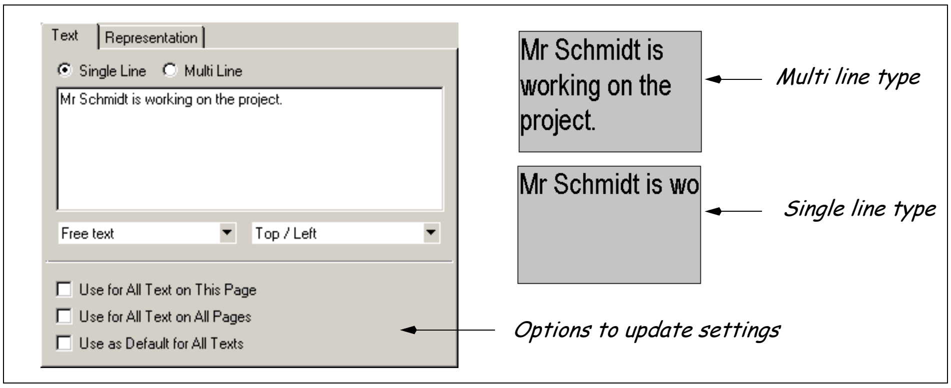
1. Choose **«Activate»** in the shortcut menu of the Object browser to switch to the presentation window.
2. Choose the PRESENTATION>TOOL>TEXTBOX command or click the text tool symbol in the Presentation toolbar.
3. The cursor becomes the symbol of the text tool. Hold the left mouse button down, draw a frame for the text and release the mouse button.
4. Double-click the frame to open the properties menu of the text element.

Type of text

Underneath the text you enter in the text frame, you can insert special text such as the date/time, page number, user name, project name, or file name.

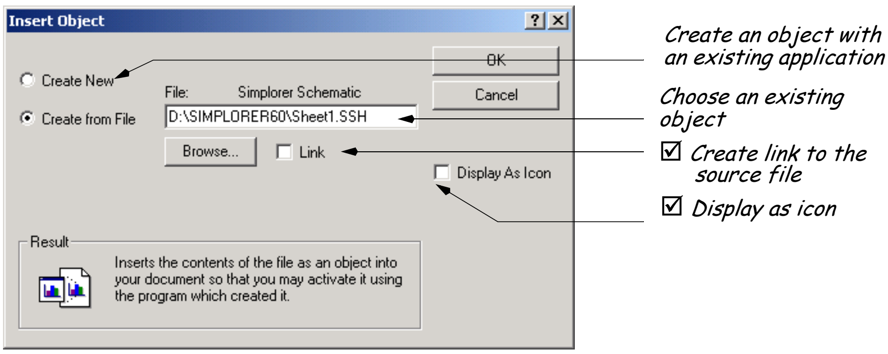
Alignment

This option sets the alignment of the text inside the frame. The properties of the text (font, color, frame) can be changed in **«Representation»**.

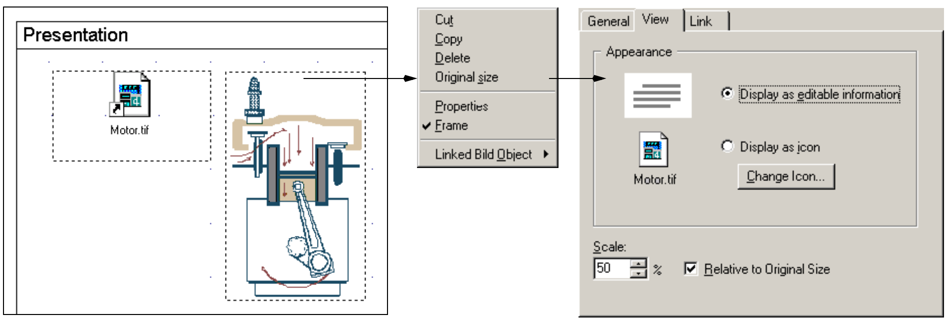


Insert OLE Objects

With EDIT>OLE OBJECTS, you can insert objects into a presentation from other Windows applications. The objects can be inserted with or without links to the original data.



The properties of OLE Objects can be changed with **«Properties»** in the shortcut menu. The **«Links»** dialog is available only if objects are linked.



You can see a list of all the links in your active presentation with EDIT>OLE OBJECTS>LINKS...

Editing Properties

You can edit or change the properties depending on the type of the element. To open the properties menu double-click the element (text), choose VIEW>PROPERTIES, or choose **«Properties»** on the shortcut menu. Double-click a graphic or a table containing simulation data to bring the window in the foreground.

General Functions for Editing Elements of a Presentation

Different functions are available to edit the element of a presentation, depending on the type of the element.

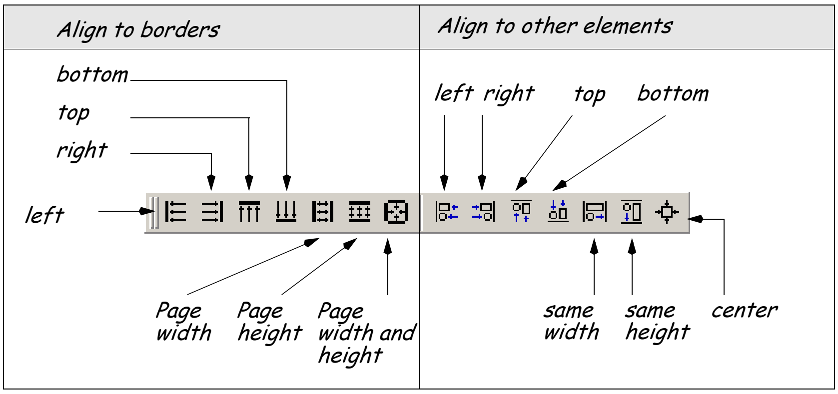
Edit Menu ▾

Command	Toolbar Symbol and Description	
CUT		Cuts the selected object and places it in the clipboard.
COPY		Copies the selected object to the clipboard.
PASTE		Inserts the contents of the clipboard.
PASTE SPECIAL		Inserts content of the clipboard in a special format.
INSERT LINK		Inserts a link.
DUPLICATE		Duplicates the selected object.
SEARCH		Searches for a certain numerical value in a table view. If the value is not available in the channel, the cursor is placed in the field with the next higher value.
SELECT ALL		Selects all objects on a presentation page.
DELETE		Deletes the selected object.
DELETE ALL		Deletes all objects on a presentation page.
OLE OBJECTS		Inserts OLE objects; shows links; changes properties of OLE objects.
OBJECT		Starts the application to edit the selected OLE object.
EDIT COORDINATE SYSTEM		Defines the coordinate system settings.
PROPERTIES SIMULATION RESULT		Displays the properties of the active data set or the simulation results.

Align Elements in a Presentation

Elements on a page can be aligned along their borders and scaled to the same height and width as the page. Elements can be aligned to each other along a chosen line or scaled to the same height and width.

All elements are scaled to the **largest** dimension of a marked element. The line of alignment always comes from the outermost element.

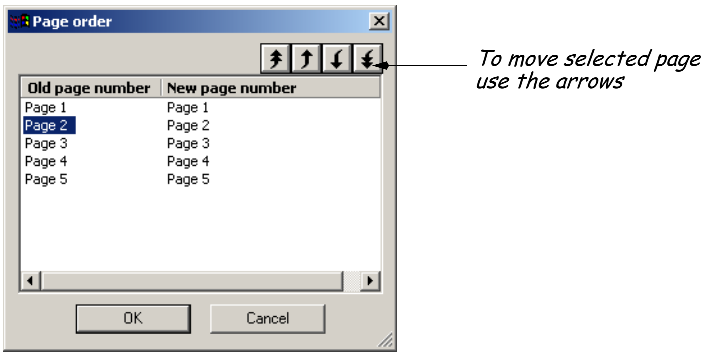


Arranging Element Order

The PRESENTATION>ARRANGE command moves a selected element one level up or down or moves the element to the foreground or background.

Add and Delete Pages

You can change number of pages with PRESENTATION>ADD PAGE and PRESENTATION>DELETE PAGE. The PRESENTATION>PAGE ORDER command organizes the pages of the presentation.



Setting the Display and Format of a Presentation

You can change the page display and format with the commands on the VIEW menu or the symbols on the Presentation toolbar. The settings affects all pages of a presentation. Grid lines and grid points are not printed.

Function	Toolbar Symbol and Description
GRID POINTS	Displays grid points.
GRID LINES	Displays grid lines.
PORTRAIT FORMAT	Defines the portrait format.
LANDSCAPE FORMAT	Defines the landscape format.

Page Layout of Presentations

You can set the screen display and page layout via the PRESENTATION>PROPERTIES command or the presentation page’s shortcut menu. Right-click on the page outside an element, and choose **«Properties»**.

The **«Page»** dialog changes the properties for the screen colors used. The **«Page frame»** dialog changes the properties for the page frame. The option **«Use for all pages»** assigns the properties to all pages of a presentation; the option **«Use as default»** assigns the properties to all new presentations.

Set Header and Footer

You can set the layout and the content of the header and footer for a presentation with **«Properties - Head line/Foot line»**.

The content of a header or footer is printed only if a position for the text is chosen, e.g., “middle” in the figure above.

Presentation Menu

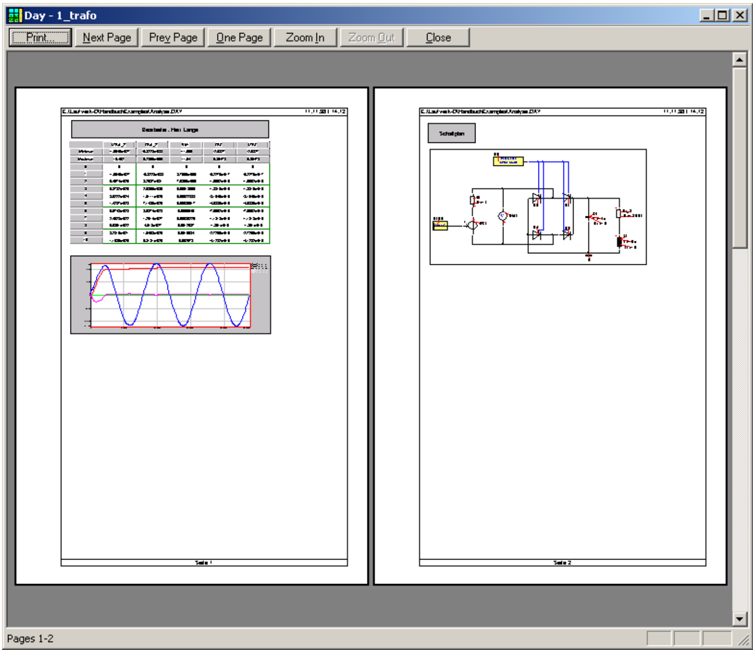
Command	Toolbar Symbol and Description
PRESENTATION NEW	Creates a new presentation.
TOOLS	Starts the tools to insert simulation results (graphics or tables) or drawing elements. See also “Inserting Elements in Presentations” on page 455.
ADD PAGE	Adds a new page to a presentation.
REMOVE PAGE	Removes the active page of a presentation.
PAGE ORDER	Changes the page order.
ARRANGE	Change the order of the elements (FOREGROUND, LEVEL UP, LEVEL DOWN, BACKGROUND); Changes properties of OLE objects
ALIGN ON MARGIN	Aligns elements along page margins.
ALIGN ON OTHER OBJECT	Aligns elements with another element (objects).
PROPERTIES	Defines settings for header, footer, and print layout.

14.1.11 Printing Simulation Data

You can print all graphics, tables, and presentations separately. The active window is printed with FILE>PRINT. You can view and change the printer settings in the FILE>PAGE SETUP menu.

Printing Graphics, Tables, and Presentations

1. To switch to the window you want to print, choose **«Activate»** on the Object browser's shortcut menu.
2. To open the preview window, choose FILE>PAGE PREVIEW, and check the settings.



3. Do one of the following:

a. Click <Print> to start printing.

b. Change the page settings:

- For graphics and tables, choose VIEW>PROPERTIES **«Print Colors»** and **«Print Page»**.
- For presentations, open the shortcut menu, define settings in the **«Page»**, **«Page frame»**, **«Head line»**, and **«Foot line»** tab.

4. If you have not already started printing, chose FILE>PRINT to do so.

14.1.12 Exporting Data

The FILE>EXPORT SIMULATION RESULTS command provides different options for exporting data in SIMPLORER file formats as well as in the formats of other applications.

Options for exporting data:

- Export all channels
- Export a single characteristic
- Export a field of characteristics

Formats of the exported data:

- SIMPLORER formats (.mdx binary and ASCII)
- ASCII formats for other applications with different separators

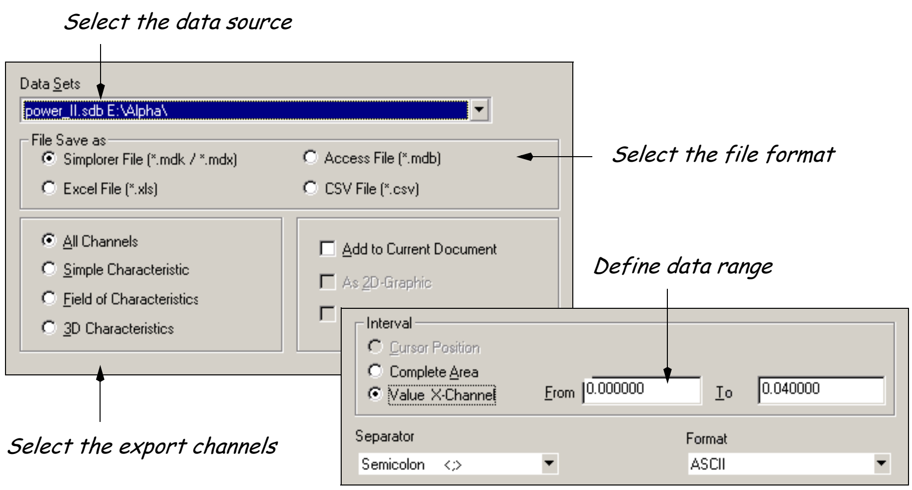
You can choose the number of samples and the range of the X channel as well.

Exporting all Data Channels of a Source File

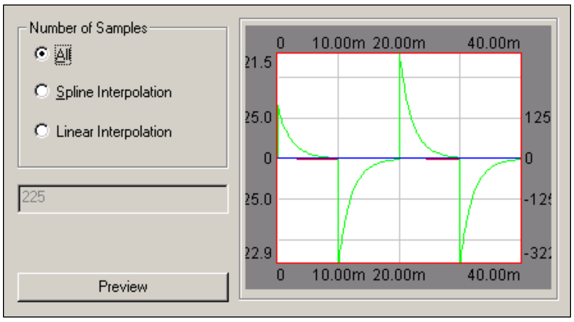
1 Choose **«Activate»** in the shortcut menu of the Object browser to switch to the analysis file containing the source data.

2 Choose FILE>EXPORT SIMULATION RESULT…

3 Select the source file (measured/simulated data) using **«All channels»**. Set the file format in the next dialog.



4 Click <Finish> and enter a file name.



Switch on preview

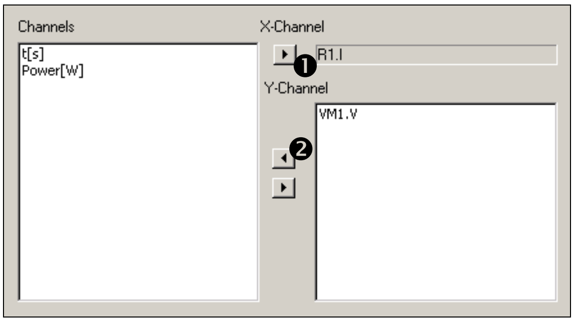
Exporting Characteristics

1 Choose **«Activate»** in the shortcut menu of the Object browser to switch to the analysis file containing the source data.

2 Choose FILE>EXPORT SIMULATION RESULT…

3 Select the source data (measured/simulated data) with **«Simple characteristic»**.

4 Set the X- and Y-channels in the next dialog.



❶ *Assign selected data-channel to X-channel*

❷ *Assign selected data-channel to Y-channel*

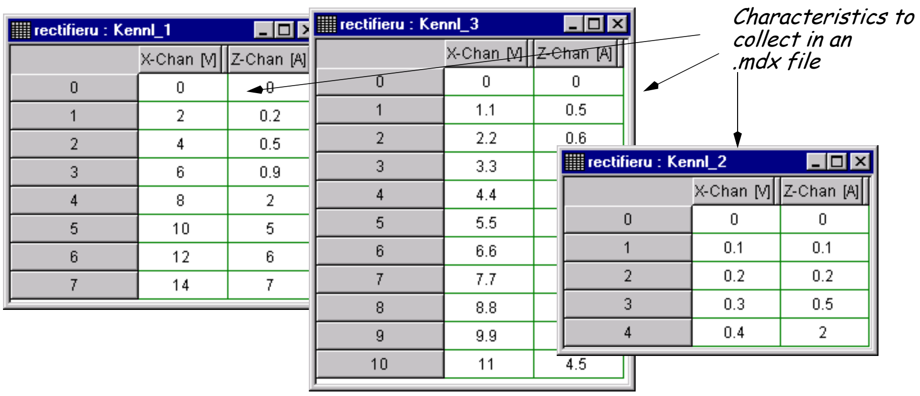
5 Select the file format in the next dialog and then the X-interval of the data.

6 Define the number of samples.

7 Click <Finish> and enter a file name.

Exporting 3D Characteristics

The functions create three-dimensional characteristics, which are used of the 3D Lookup Table component to describe, for example, the behavior of a temperature-dependent resistor. The Export 3D characteristics function creates a special .mdx file, which represents a parameter-dependent family of characteristics.



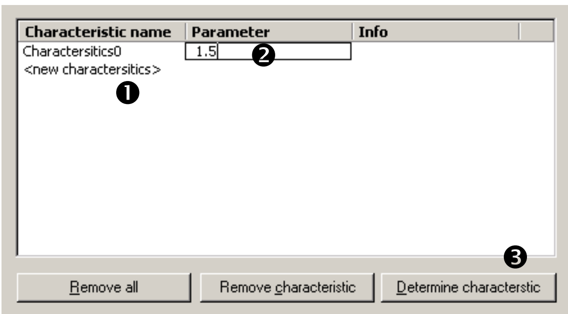
1 Choose «**Activate**» in the shortcut menu of the Object browser to switch to the analysis file containing the source data.

2 Choose FILE>EXPORT SIMULATION RESULT.

3 Select the option «**3D characteristics**» and click <Continue>.

4 Double-click «**New characteristics**» to create a new entry.

Create in step **4** the characteristics which should be saved in one file and assign in step **5** the corresponding value for the family parameter.

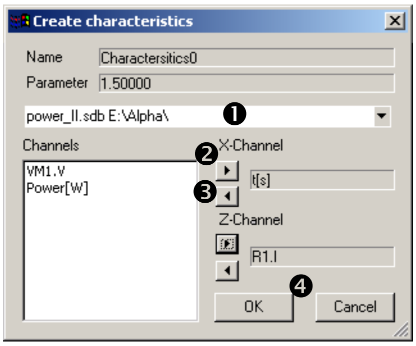


❶ *Double-click «New characteristics»*

❷ *Enter the value for selecting the characteristic*

❸ *Determine data channels for characteristic*

5 Select the data set from the list. Define the data channels for the X and Z axes.



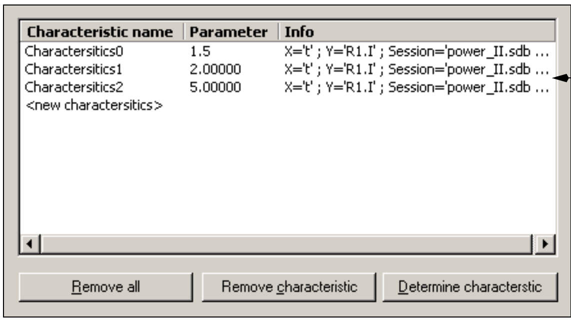
❶ *Select data set (simulation run)*

❷ *Apply the selected data channel for X-channel*

❸ *Apply the selected data channel for Z-channel*

❹ *Click <OK> to apply the settings*

Repeat steps **4** and **5** until all characteristics have the family parameter, the X and the Z axes.

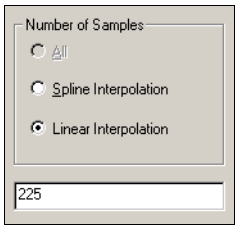


Defined family of characteristics

6 Define the file format and the range of the X axis of data in the next dialog. The maximum range is set as default.

7 Define the number of samples. The maximum number existing in one characteristic is set as default. Select the form of interpolation: Spline- (curve) or linear interpolation. Click <Finish>.

8 Enter a file name for the characteristic file and click <Save>.



All characteristics within the file have the same number of samples. The value of the family parameter has to monotonous increase across the file.

Using the Clipboard

Use EDIT>COPY to copy graphics and selected elements from presentations into the Clipboard; you can then paste and insert them into other Windows applications.

14.1.13 Arranging Screen Layout and Using Help

Online help and version information are both available with the DAY Post Processor, as for all other SIMPLORER programs, from the HELP menu.

Arranging Windows

In the DAY Post Processor several files can be edited at the same time in different windows. The arrangement of the windows on the screen is managed with the window commands, accessible from the symbols on the toolbar or the WINDOW menu.

Window Menu ▾

Command	Description
NEW	Opens a new empty window.
OVERLAPPING	Rearranges all open windows in an overlapping manner (cascade). All title bars are visible.
TILE VERTICALLY	Rearranges all open windows side by side.
TILE HORIZONTALLY	Rearranges all open windows one on top of the other.
ARRANGE ICONS	Moves all minimized windows to the lower part of edit window and arranges them.
«OPENED WINDOWS»	Lists all open Windows and the corresponding file names.

Help and Version Information

Online help is available for all SIMPLORER programs, from the HELP menu.

Help Menu ▾

Command	Description
HELP CONTENT	Go to online help.
ABOUT DAY	Product and version information.

14.2 SIMPLORER ASCII System Data Formats

- .mdx files (.mtx)
- .mdk/.mda files
- External Formats

The .mdx File Format

The .mdx file is a SIMPLORER intern data format to exchange data between the different programs. You can create and edit the files with a common Text Editor. The semicolon at the end of each data set is essential, including the last one.

Data files in .mdx format are structured in a table-like fashion. The data sets are interpreted as a sequence of k-dimensional value tuples (ai,bi,ci,….) with a record length n (i.e., data field/table of k columns times n rows). The correlated values of a k-tuple are arranged successively in the data set:

```
(a1,b1,…,z1,    a2,b2,…,z2, … ,an,bn,…,zn)
\________/      \_______/ …      \_______/
1st k-tuple       2nd k-tuple       nth k-tuple
```

In a typical case (time functions, characteristics) the first value in the n-tuple (ai) should represent the common abscissa (“x”, time “t”…) where the data sets should be in ascending order of ai ("x", "t", …).

Each file starts with header information for the data sets, which directly follows the “END:=” statement. All data within a line are separated by a semicolon; a line must be closed by a semicolon and CR/LF.

```
TYP:=A
DIM:=3
LEN:=10
N1:=C2.V
U1:=V
N2:=Filter
U2:=
N3:=x
U3:=
END:=
0;              0.00001600;     0.00000400;
0.00001587;     0.00004787;     0.00001998;
0.00003168;     0.00009548;     0.00005188;
0.00006323;     0.00015872;     0.00009959;
0.00011041;     0.00023744;     0.00016299;
0.00019170;     0.00033153;     0.00024193;
0.00026967;     0.00044087;     0.00033631;
0.00036288;     0.00056532;     0.00044599;
0.00047123;     0.00070470;     0.00057084;
0.00061599;     0.00085909;     0.00071075;
```

The specifications Nxx and Uxx for names and units of data channels are optional and can be omitted.

Data Set Parameters

The Required Parameters are (in this order):

TYP:=	data file type • 4 4-Byte-Real data; The data sets are read only via the separate conversion program. • A ASCII data
DIM:=	dimension n of the file (number of columns of the table) Usually in a simulation data file this corresponds to the number of data channels
LEN:=	number of data records (for ASCII files = number of rows) In a simulation data file this is equal to the number of time steps calculated.
END:=	close of data identification set

The Optional Parameters (in arbitrary sequence and selection) are:

SEP:=	separator between the single data values of ASCII sets; this specification is necessary only when selecting a separator different from the default semicolon character (";"). The separator may be enclosed by quotation marks (" "), e.g. SEP:=/ or SEP:="/"
COD:=	kind of data (3 Characters)
DAT	time domain data
NAM:=	data file name (max. 64 characters), e.g., used for titles in graphics
USR:=	user name (max. 64 characters allowed)
NEW:=	date of creation (DD.MM.YY)
MOD:=	date of latest modification
TIM:=	time of creation (HH:MM:SS)
Nxx:=	name of the XXth quantity in the data record (the first 14 characters are considered to be valid in DAY); can be used, e.g., for automatic axes labeling; XX = '01' … n (n - data set dimension); leading zeros must be specified for XX.
Uxx:=	unit of the XXth quantity in the data record - analogous to 'nXX' (max. 6 characters valid); may be SI unit without prefix - the data are supposed to be in the record in SI units and unscaled; otherwise, confusion may result in graphic representations.

Optional supplementary parameters, especially for measured transient recorder data:

DX2:=	2nd sampling rate when dual time base is used

There are much more possible optional parameters in the .mdx file header. The experiment tool creates .mdx files for simulations results of experiments. To get these parameters open such a .mdx file with the Text Editor. The header is set always with “END:=”.

Data Set Format

- n-dimensional value sets are possible. Each record containing n values has to be positioned on one line (limited to 255 ASCII characters per record).
- The k values are separated by a separator character (also after the last value per line!) usually the semicolon (;) - space characters are then ignored.
- All other special ASCII characters are also possible separators, except for the decimal point (.) and the space.
- If the separator is not the semicolon, the separator must be specified in the data set identification file as the SEP:= parameter, e.g., SEP:="/".
- The data can be written in any standard number format used in data processing, e.g., 10/ 10.0/ 1.0E1.

.mdx Files used of Components using 2D Characteristics

```
typ:=A
dim:=2
len:=6
end:=
0;0;
0.2;0.01;
0.5;0.02;
0.8;0.03;
1;2;
1.5;200;
```

.mdx Files used of Fourier Sources

```
typ:=A
dim:=3
len:=6
n01:=f
u01:=Hz
n02:=Amplitude
n03:=Phase
u03:=rad
end:=
50;0;0;
50;1;0;
100;2;0;
150;3;0;
200;4;0;
250;5;0;
```

.mdx Files used of 3D Lookup Tables

```
typ:=A
dim:=3
len:=12
n01:=V
n02:=T
n03:=A
end:=
0;0;0;
1;1;0;
2;2;0;
0;0;0.2;
1;2;0.2;
2;4;0.2;
0;0;1;
1;4;1;
2;6;1;
0;0;1.2;
1;6;1.2;
2;8;1.2;
```

.mdx Files used of DES Models

```
Ord=2
Dim=2

Name=DES1

M_0_0_0=C1
M_0_0_1=-C1
M_0_1_0=-C1
M_0_1_1=C1

M_1_0_0=Ki1
M_1_0_1=-Ki1
M_1_1_0=-Ki1
M_1_1_1=Ki1

M_2_0_0=J1
M_2_1_0=J3
M_2_1_1=J2

RS_0=Jump1
RS_1=0

IC_0_0=5
IC_0_1=4
IC_1_0=-4
IC_1_1=-5
```


.mdk/.mda File Format

Earlier SIMPLORER versions used the .mdk/.mda file format to provide data in ASCII format for simulation models or the DAY Post Processor. Both files must have identical names and be placed in the same directory. The .mdk file (the identification file for the data set) contains only the header information, corresponding to the file-structure definitions described for the .mdx format. The .mda file contains the data sets. The .mdx file specifications are also valid.

Header in the .mdk file:

```
typ:=A
dim:=2
len:=10
end:=
```

Data sets in the .mda file:

```
0.0;   0.0;
0.1;   0.587;
0.2;   0.951;
0.3;   0.951;
0.4;   0.587;
0.5;   0.0;
0.6;   -0.587;
0.7;   -0.951;
0.8;   -0.951;
0.9;   -0.587;
```

External Formats

If you want to use external data formats in characteristics or data evaluations, they must correspond to the standards (.csv, .mdb, .csv). In some cases, the interpretation depends on the system settings on your PC. Excel files must be saved in the 4.0 format and may contain only one table. At the beginning of the files, one line is reserved for channel names.

Header and data sets in ASCII data format:

```
"Ch1","Ch2"
1.00,2.00
3.00,4.00
5.00,6.00
7.00,8.00
```

14.3 DAY Optim Post Processor

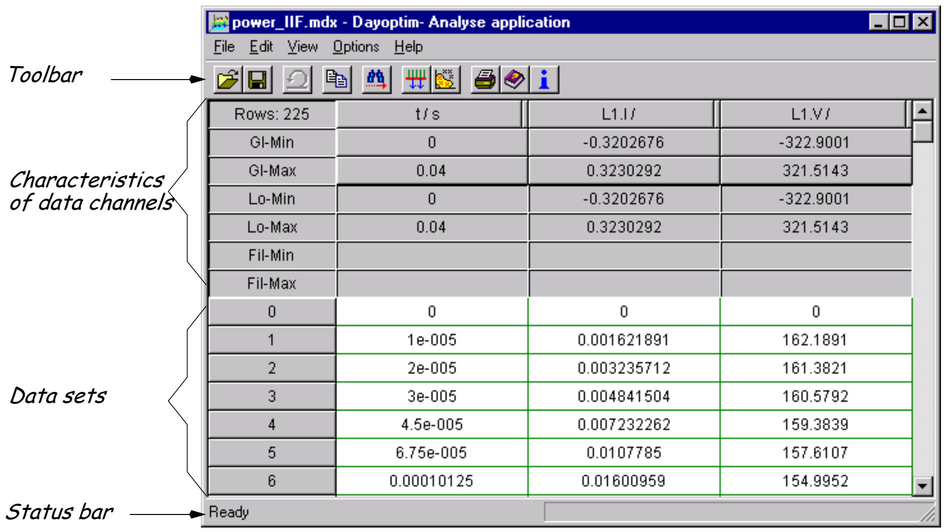
DAY Optim is a program where you can evaluate results from the SIMPLORER Experiment Tool. With DAY Optim, multidimensional data sets, usually created of experiments, can be read, processed, filtered, and sorted. You can determine local and global minimal and maximal values, which can be used for parameters in the analyzed simulation model later.

From the SSC Commander, double-click the DAY Optim symbol or choose PROGRAMS>DAY OPTIM to start the DAY Optim.

<u>This section contains information on</u>

- DAY Optim main window
- Parameters of the data channels
- File management
- Modifying the presentation (sort, copy, paste)
- Changing the settings

DAY Optim Main Window



The settings for the elements of the main window can be changed in the OPTIONS>VIEW menu.

Parameters for Data Channels

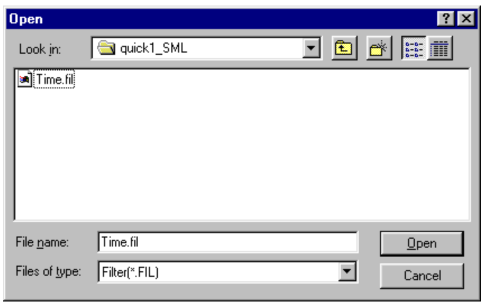
In the header rows of the table essential parameters and values of the data channel are summarized. The first line of the table shows the name and unit of the data channel and the first column shows the line number. The first field contains the sum total of data channels.

Parameter	Description
Gl-Min	global minimum of data channel for all data sets of the file
Gl-Max	global maximum of data channel for all data sets of the file
Lo-Min	local minimum of data channel for remaining data sets (after using filters and deleting data sets)
Lo-Max	Local maximum of data channel for remaining data sets (after using filters and deleting data sets).
Fil-Min	Lower limit of the filter used.
Fil-Max	Upper limit of the filter used.

File Management in DAY Optim

File Menu ▾

Command	Toolbar Symbol and Description	
NEW		Creates a new file.
OPEN		Opens an existing .mdx file
SAVE		Saves .mdx files.
SAVE AS		Saves the file with a new name.
EXPORT FILTER		Save the active filter used for data channels into a file. The filter is set in the EDIT>FILTER menu.
IMPORT FILTER		Imports a filter file (.fil).



Dialog to open and apply an existing filter file

PRINT		Prints the data sets.

Command	Toolbar Symbol and Description
PRINT PREVIEW	Shows the print page. Settings for print layout are saved in the menu OPTIONS **«Print Header»** and **«Print Footer»**.
PRINT SETUP	Opens the Windows dialog for printer settings.
«LAST OPENED FILES»	Shows a list of the last opened data analysis files.
EXIT	Closes the DAY Optim.

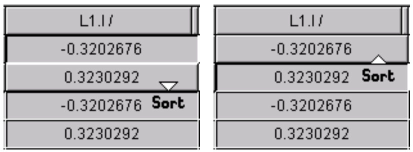
Modifying the Presentation

Selection of data channels and data sets

Click the column head or the line number to select the entire data channel or the entire data set. The selected area is highlighted in yellow. Click again to cancel the selection.

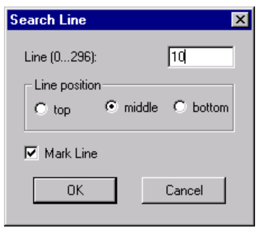
Sorting data channels

The mouse pointer changes its shape when sliding over the global minimum or global maximum areas of a data channel. When the mouse pointer takes the shape of an arrow, click to sort the selected data set in ascending or descending order. The right picture shows sorting in ascending and the left in descending order.



Edit Menu

Command	Toolbar Symbol and Description
UNDO ALL	Undo all changes.
COPY	Copy selected areas into the clipboard. The contents of the clipboard may be used by any Windows application.
SEARCH	Enter the line number to which the display is scrolled. If the option **«Mark line»** is checked, then the searched line is also marked.

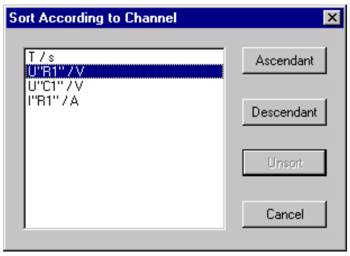


Definition of line number

Command | **Toolbar Symbol and Description**

SORT

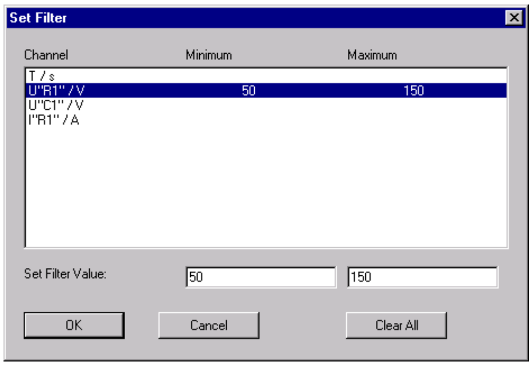
All data channels are sorted according to the selected data channels in ascending or descending order.



Set data channel and sort option

FILTER

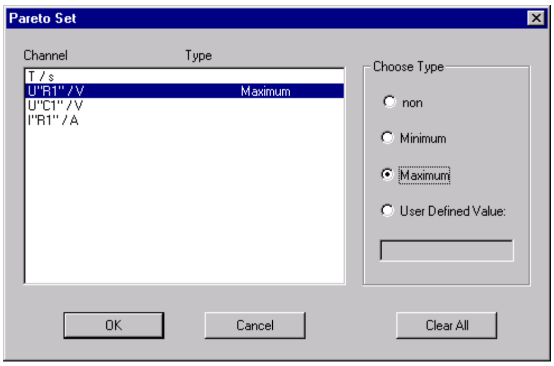
The filter function can be used to search for all values of a data set which match the interval specified by the Minimum and Maximum parameters.



Define the maximum and minimum value

PARETO SET

Creates a Pareto set. A data set belongs to the Pareto set if at least one of its criteria (“type”) is better than that of all other data sets. The type must be set for each selected data channel.

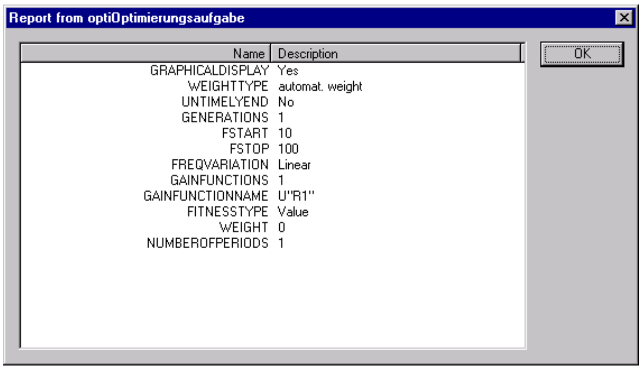


Criterion for the calculation of the Pareto set

View Settings and Options

View Menu ▾

Command	Description
REPORT	Shows the general description of an experiment.

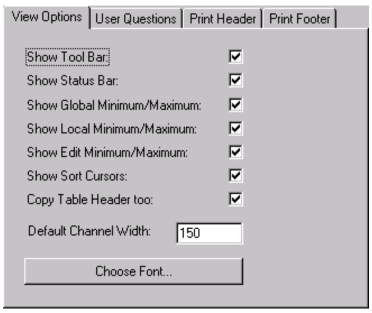


Data window of an optimization task

Command	Description
TOOLBAR	Switches the display of the toolbar ON or OFF.
STATUS BAR	Switches the display of the status bar ON or OFF.
DEFAULT WIDTH	Sets the column width to the default values defined in OPTIONS «View Options».

Option Menu ▾

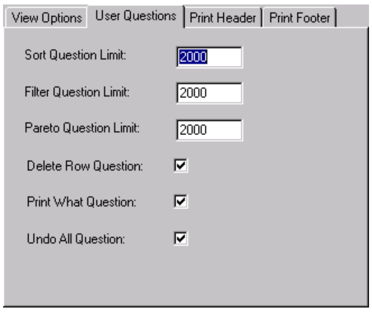
View Options



Set elements for screen presentation

This menu sets the appearance of DAY Optim. The value <Default Channel Width> sets the width of columns when a file is opened. The width can also be manually adjusted by dragging the column borders. The font used is set with <Choose Font>.

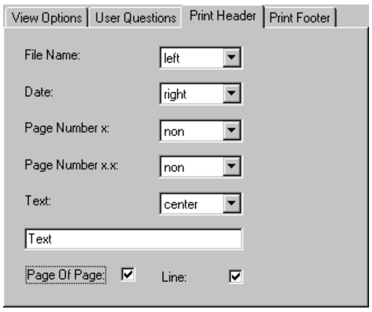
User Questions



Set options for user questions

If the size of an existing data set exceeds a determined value, then user confirmation is required for sorting, filtering, and computation of parameters. This helps to save time if the data sets are large. If the options **«Delete Row Question»**, **«Print What Question»** or **«Undo All Question»** are checked, then a confirmation is requested before these operations.

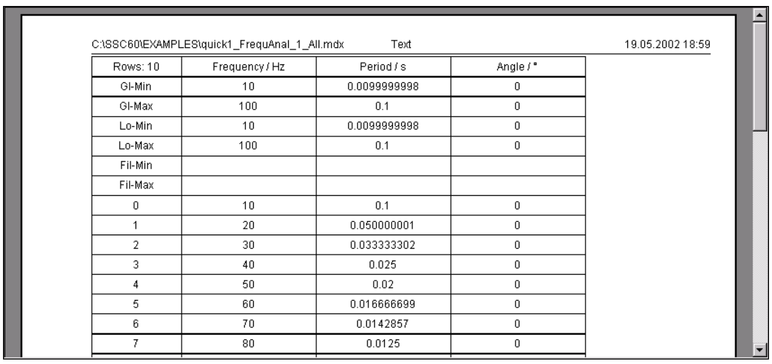
Printing Header and Footer



Select a location of header and footer elements

The PRINT HEADER/PRINT FOOTER menus contain options for the layout of headers and footers. In the Drop Down Menu you can define the properties of headers or footers: **«none, right, left, or center»**. The text appears only when a text position is given (e.g., "center" in the figure). The option **«Page Of Page»** prints the active and the total page numbers at the header or footer position defined above.

The content of a header or footer is printed only if a position for the text is chosen.



C:\SSC60\EXAMPLES\quick1_FrequAnal_1_All.mdx Text 19.05.2002 18:59

Rows: 10	Frequency / Hz	Period / s	Angle / °
Gl-Min	10	0.0099999998	0
Gl-Max	100	0.1	0
Lo-Min	10	0.0099999998	0
Lo-Max	100	0.1	0
Fil-Min			
Fil-Max			
0	10	0.1	0
1	20	0.050000001	0
2	30	0.033333302	0
3	40	0.025	0
4	50	0.02	0
5	60	0.016666699	0
6	70	0.0142857	0
7	80	0.0125	0

Print preview of data file

Help and Version Information

Online help is available for all SIMPLORER programs, from the HELP menu or by clicking the symbol on the toolbar.

Help Menu ▾

Command	Toolbar Symbol and Description	
HELP CONTENTS		Go to online help.
ABOUT DAY OPTIM		Product and version information.

15 Model Libraries

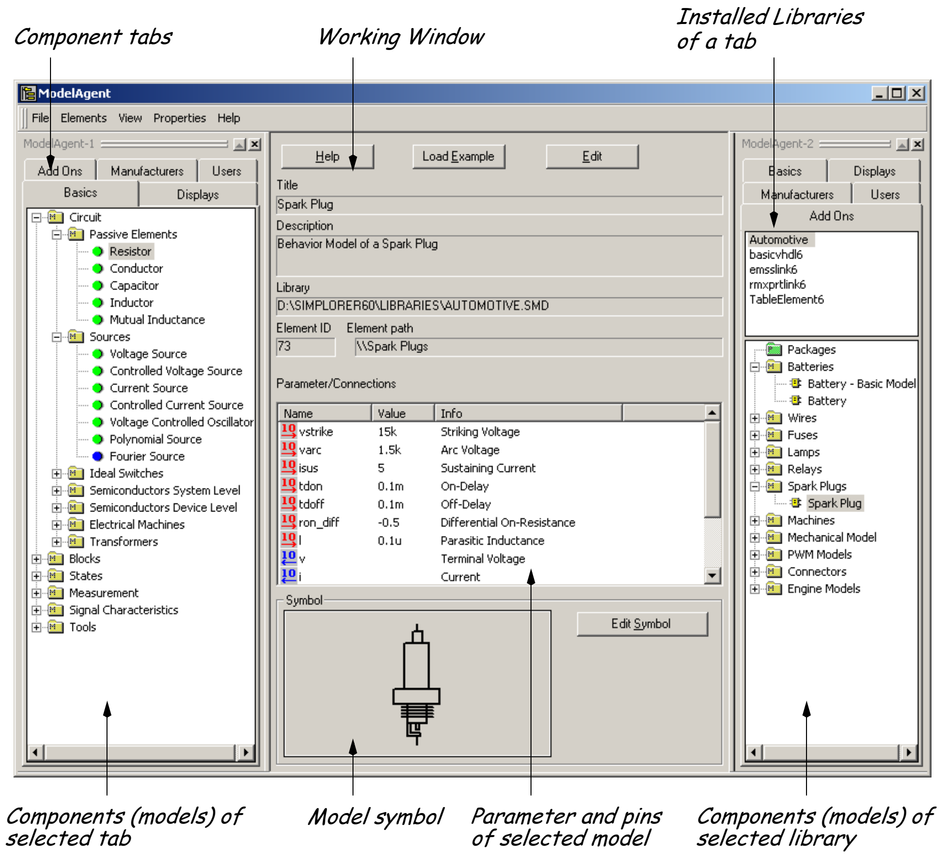
SIMPLORER arranges models (or components) in model libraries. The Model Agent maintains all libraries and models and provides editing functions, which are the same as the Model Agent functions in the Schematic. To start the Model Agent from within the SSC Commander, double-click the Model Agent symbol  , or choose PROGRAMS>MODEL AGENT.

This chapter contains information on:

- Viewing model libraries with the Model Agent
- Creating model libraries
- Converting libraries
- Copying, moving, and deleting models
- Searching models
- Look and encode models
- Creating new models in a model library
- Model properties

15.1 Model Agent Main Window

When you first start the Model Agent, a window similar to the one below appears. Model trees contain the SIMPLORER installed libraries and their corresponding models.



If you selected different VIEW menu options, then your window may look different than the figure shown above. To get the same display as shown, enable all options on the VIEW menu, and maximize the window.

Two windows display the structure of the installed libraries and the models included in each. When you select a library, its model tree appear in the lower-right window. If you move the mouse pointer over the model name, a Tooltip appears explaining the symbol.

The working window displays the parameters and connections for the selected model. From this window, you can create or modify the macro definition. To change a symbol, use the <Edit Symbol> command to open the Symbol Editor.

15.2 Library Management

In SIMPLORER, you can create your own libraries and insert both user-defined and existing models. Each model is placed in a library, which must be included in the SIMPLORER environment so that it can be used in a simulation. Use the FILE menu commands to install and configure the libraries in the SIMPLORER program environment. Use the ELEMENT menu commands to arrange the models within a library.

Managing Model Libraries

All model libraries used in SIMPLORER are .smd files. Each library must be included in the SIMPLORER environment.

You can create, add, and remove libraries from the following locations:

- Model Agent FILE menu
- Model tree shortcut menu (in the Model Agent, Schematic, and Symbol Editor)
- SSC Commander OPTIONS>MODEL DATABASE (without create)

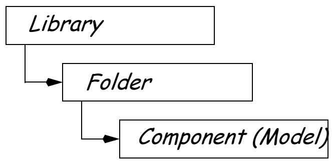
You can select a tab (**«Basics»**, **«Displays»**, **«AMS»**, **«Digital»**, **«Tools»**, **«Add Ons»**, **«Manufacturers»**, **«Users»**, **«Project»**) where the libraries should be included. Libraries assigned to the project tab are available only for the corresponding project. If you want to use the library in other projects, you have to include it again or assign it to other tabs.

File Menu ▾

Command	Description
INSERT LIBRARY	Inserts an exiting library (.smd) into the model tree. If a 5.0 library is inserted, a prompt to synchronize the file appears.
NEW LIBRARY	Creates a new model library. Select the language resources and create a new library by choosing <OK>. See also “You can create content folders to structure the content within a libary. These content folders make it easier to search for large libraries. Use the Elements>New> Folder menu command to install a new folder. You can change the name immediately or later by clicking twice in the name field and entering a new name.Creating Libraries” on page 482.
REMOVE LIBRARY	Removes a library from the model tree (but not from the computer system).
SIMPLORER MODELLING IMPORT	Imports .smu model files from the SIMPLORER Web Database. See also “Importing .smu Model Files of the SIMPLORER Web Database” on page 484.
CREATE INTERFACE DESCRIPTION	Creates a text description of all models used in the selected library and their parameters, exporting a table containing the following fields: Name, Parameter, Direction, Dynamic/Vector, Output, Default, Unit, Data Type, Node, Nature, **«Language Resources»**, Path in library.

Command	Description
COMPACT AND REPAIR LIBRARY	Compacts and repairs a SIMPLORER model library. Enter the name of the library, and start the process. Before you can start the repair process, you have to unistall the corresponding model library. The model library must not be installed.
CONVERT 4.X LIBRARY	Converts an existing SIMPLORER 4.x library into a SIMPLORER library 6.0. See also “Converting 4.x Libraries” on page 485.
EDIT STRINGS	Displays all text strings used from the selected library, and allows you to modify entries. The order of the strings is irrelevant to the models.
EXPORT LIBRARY	Creates a separate .sml or .vhd file from the selected model library with all model descriptions, either in SML text (choose SML20) or VHDL-AMS (choose VHDL-AMS).
PROPERTIES	Displays all existing language tables from the selected library (text and modeling language). You can add a new resource with <New>. See also “SIMPLORER Language Concept” on page 483.
EXIT	Closes the Model Agent.

You can create content folders to structure the content within a libary. These content folders make it easier to search for large libraries. Use the ELEMENTS>NEW> FOLDER menu command to install a new folder. You can change the name immediately or later by clicking twice in the name field and entering a new name.



Creating Libraries

To create a new library do the following:

1. Choose FILE>NEW LIBRARY in the Model Agent or choose **«New Library»** on the shortcut menu of a tab in the Model Agent, Schematic, or Symbol Editor.
2. Define library name and location. Click <Save>.
3. Click <New> in the **«Resource Tables»** area, and select a language for model text strings. Parameter names are identical for all language resources.
4. Click <OK>. A new set of entries appears in the list.
5. Repeat steps **3** and **4** if you need more than one language.
6. Click <New> in the **«Simulation Tables»** area, and select SML20 for the simulation description used in the version 6.0.
7. Click <OK>. A new set of entries appears in the list.
8. Repeat steps **6** and **7** if you need also VHDL-AMS as modeling language.
9. Click <OK> to create the new model library.

The model library is inserted into the model tree without a model. You can insert models via ELEMENT>NEW or by dragging and dropping from other libraries. The name for the model library and language resources can be modified later.

SIMPLORER Language Concept

SIMPLORER supports different languages for the following items:

- Program menu and dialogs
- Model libraries.

The language for program menus and dialogs is defined in the SSC Commander, via OPTIONS/ LANGUAGE. Select **«German (Germany)»** or **«English (English)»**, and click <OK>. After an additional dialog box, if you really want to change this setting, SIMPLORER closes and restarts with the new language for menus and dialogs. Before the process, all opened documents need to be saved and closed.

The language for model libraries is defined when a new library is created or with FILE>PROPERTIES in the Model Agent for an existing library.

To add a language resource to an existing library do the following:

1 Select the library you want. Choose FILE>PROPERTIES in the Model Agent or choose **«Properties»** on the shortcut menu of a tab in the Model Agent, Schematic, or Symbol Editor.

2 Click <New> in the **«Resource Tables»** area, and select a language for model text strings. Parameter names are identical for all language resources.

a. Select **«Copy Resource»** to copy data from an existing language.

b. Select one of the existing language from the list.

c. Select Symbols, Keywords, Infos, and/or Files settings and click <OK>.

3 Click <OK>. A new set of entries appears in the list.

4 Click <OK> to save the settings.

The new language resource is available for all models in the library. If you do not copy a languages resources, you have no entry in title field. For the other settings default settings are created of the Model Agent.

You can set library languages from the following locations:

- SSC Commander, via OPTIONS>LANGUAGE **«Library Language»** (for Model names displayed in the Model Tree)
- Single Schematic sheet, via SHEET>PROPERTIES **«System»«Language»** (for names and text used of models placed on the sheet). **«System dependent»** means the language defined in the SSC Commander Library Language will be used.

The table shows available language settings for both, menus and dialogs, and model libraries and the effect on SIMPLORER.

	Menu and Dialogs	**Model Libraries**	
Settings	SSC Commander: OPTIONS>LANGUAGE **«SIMPLORER Language»** (German or English)	SSC Commander: OPTIONS>LANGUAGE **«Library Language»**	Schematic: SHEET>PROPERTIES **«System»«Language»**
Effect on	Program menus and text in dialogs of all SIMPLORER programs	Names in Model Trees	Names and strings in a single Schematic sheet

Importing .smu Model Files of the SIMPLORER Web Database

The model Web Database provides SIMPLORER semiconductor simulation models. On the web, you can search for different criteria, download the models you select, and install the models in new or existing SIMPLORER libraries.

The Web Database models can only be used if you have purchased the optional SIMPLORER SPICE-Pack. For more information, please contact your Ansoft sales representative or the SIMPLORER sales team at sales@simplorer.com.

Search Models in the Web Database

You can either search for models within groups (BJT, Diode, MOSFET, JFET, OPAMP, or IGBT) or define a search term (Model names, Model types, Manufacturers). If you do not want to enter a specific model name, you can search for numerical value ranges for typical parameters.

If you do not specify a value, the complete model list of the selected type and manufacturer is displayed. If no model meets your specification, you can send an e-mail inquiry to Ansoft, and ask for the model with these specifications.

Information about Models

If you select the model name in a list, you get the following information:

- Model name
- Program requirements
- Example sheet
- Data sheet

You can also add a model to the download list. Click <Back> to change to the last search result.

Download Selected Models

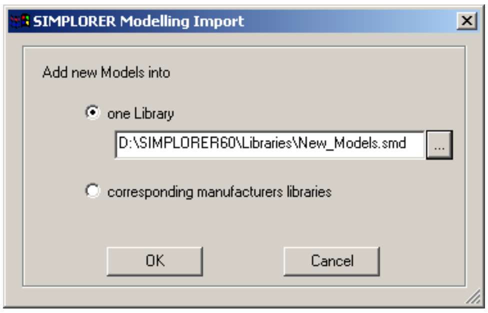
1. Select the box **«Add to download»** to add a model to the download list. You can add models from different model types and manufacturers in one list.
2. Click the disk symbol on the upper-right of the table.
3. Click <Continue>.
4. Select the box to confirm the license agreement.
5. Select a directory for the model file *mylibrary.smu*, and click <OK> to start the download. Select *All Files* in the file type list.

Installing Downloaded Models

Models downloaded from the Web Database must be installed in SIMPLORER.

1. Click FILE>SIMPLORER MODELLING IMPORT.
2. Select the file *mylibrary.smu* that contains the desired SIMPLORER models.
3. Do one of the following:

a. Select **«One Library»** to create or open a single library for the new models:



1. Click the <...> button to create or open a library for the new models.
2. Enter a new name to create a new library, or browse for a .smd file to open an existing library.
3. Click <OK> to install the new models.

The library with the defined name is inserted in the in **«User»** tab of the Model Agent.

b. Select **«Corresponding Manufacturer Libraries»** to install multiple libraries:

1 Click <OK> to install the new models. The libraries with the name *manufacturer_name.smd* are inserted in the in **«Manufacturers»** tab of the Model Agent.

Converting 4.x Libraries

Libraries from former versions of SIMPLORER (4.x) cannot be used in version 6.0 without being converted.

To convert a library from a previous version:

1 Click FILES>CONVERT 4.X LIBRARY.

2 Select the old library, and define a new name and location.

3 Specify whether the library is only for your own local use (**«Local»**) or for all users on a network (**«Public»**). When **«Public»** is selected, mapping tables are shared. When **«Local» is selected,** an additional dialog box opens, asking you to enter the path for the mapping file.

4 If an ambiguous model definition causes error messages during the conversion process, do one of the following:

a. To insert the model in the new library anyway, click **«Insert Element»** or **«Insert all Elements»**.

b. To leave that model out of the new library, click **«Skip Element»** or **«Skip all Elements»**.

5 To edit the model text, click <Edit>. Model text can only be edited if the SML text is not encoded.

6 Click <OK>.

When you start a second conversion run for a library, the created mapping information of the first run is used.

The successful conversion depends on the settings in the original SML text of the 4.x library. Please contact SIMPLORER support if problems occur.

15.3 Component Management

You can use SIMPLORER to create your own models. Each model is placed in a library, which must be included in the SIMPLORER environment so that you can use the model in a simulation. Use the ELEMENT menu commands to create, insert, modify, and arrange models within a library.

Most of the commands described in this chapter are available only if a component is not locked (ELEMENTS>LOCK ELEMENT) and if the model library file is not write protected (usually e-mailed files).

Copying, Moving, and Deleting Components (Models)

Models can be reorganized within the libraries as follows:

Copy By holding down the CTRL key the selected model can be dragged to another place within the model tree (ELEMENTS>COPY).

Move By holding down the SHIFT key the selected model can be dragged to a new place in the within the model tree (ELEMENTS>PASTE or PASTE ABOVE or PASTE IN FOLDER).

Delete Choose ELEMENTS>REMOVE ELEMENT and apply with <Yes>

Deleted components cannot be restored. Should you lose models in this way, you will need to reinstall SIMLPLORER in order to restore them.

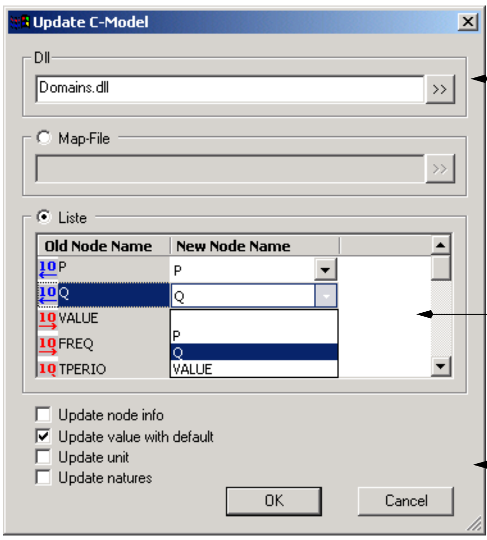
Element Menu ▾

Command	Description
NEW	Creates a new folder or macro, or imports a C model from a .dll file. See also "Creating Your Own Models (Components)" on page 489.
INSERT	Imports SML or VHDL-AMS macros from files.
EXPORT	Creates a separate .sml or .vhd file from the selected model, either in SML text (choose SML20 MACRO) or VHDL-AMS (choose VHDL-AMS MACRO).
REMOVE ELEMENT	Deletes the selected element **permanently**. The element cannot be installed again.
EDIT SYMBOL	Opens the symbol editor to modify the symbol for element. See also 16 "Symbol Editor" on page 499.
UPDATE C MODEL	Starts a update for the selected C model. See also "Updating C Models" on page 487.
COPY	Copies the selected model.
PASTE	Inserts the model below the selected model in the tree.
PASTE ABOVE	Inserts the model above the selected model in the tree.
PASTE IN FOLDER	Inserts the model in the selected folder in the tree.

Command	Description
HELP FOR ELEMENT	Opens the help file for the selected model.
EXAMPLE	Opens the example file for the selected model.
LOCK ELEMENT	Locks the selected component in the model tree. You cannot make any changes again if a component is locked. To remove the lock you have to copy the component.
PROPERTIES	Opens the input dialog to edit the macro content.

Updating C Models

To update models using .dll files, either click ELEMENTS>UPDATE C MODELS, or use the shortcut menu command. Select a .dll file in the input field. The model name in the .dll file must be identical to the active model name (defined with RegisterUserModel). You can select either a map file or the list mode to assign new to old parameters. Click in a field of new parameters to select a new name in the list. The list provides all in the C model defined parameters and nodes.



The options to update node model information, default values, and units replaces entries in the active model with definitions from the .dll file. Units are defined with *SetUnitNameNode_nc*, information about nodes with *SetInfoNode_nc* and *SetInfoNode__c*, default values in *AddNode_nc*. See also C Interface in the online help.

Inserting Models in SML Text

Every model in the Model Agent can be dragged with the mouse directly into the Text Editor. If the Text Editor option SETTINGS>OPTIONS>GENERAL OPTIONS **«Add comment»** is active, the general description is copied in the text, along with the model's macro call. Errors can be reduced by using the **«Add comment»** feature, especially for extensive macro calls.

Symbols in the Model Agent

	Red triangle. Elements for displaying simulation results directly on the sheet in the Schematic. You cannot use results of the these elements as parameters for the simulation model.
	Green point. Internal SIMPLORER components. Most of Basic components are internal ones. The models are calculated within the internal simulator.
	Blue point. Standard and user-defined C models. There are a few components of this type in the **«Basics»** tab. These models are based on the C interface of the internal simulator.
	Yellow rectangle. Standard and user-defined text macros. You can create macros in the Model Agent in SML or VHDL-AMS text, edit, and encode. If you encode a macro script, you cannot edit it again. A text macro on the sheet cannot be opened.
	Light-blue rectangle. Standard and user-defined graphic macros. You can create macros in the Schematic from a graphical selection. A graphic macro on the sheet (subsheet) can be opened.
	Green rectangle. VHDL-AMS definitions (packages). The element cannot placed in the Schematic.
	Green rhomb. Coupling components. The models are calculated by different simulators.

You can found the component type of already dropped models in their property dialog in the **«Library»** tab.

Modifying Symbols

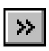
15

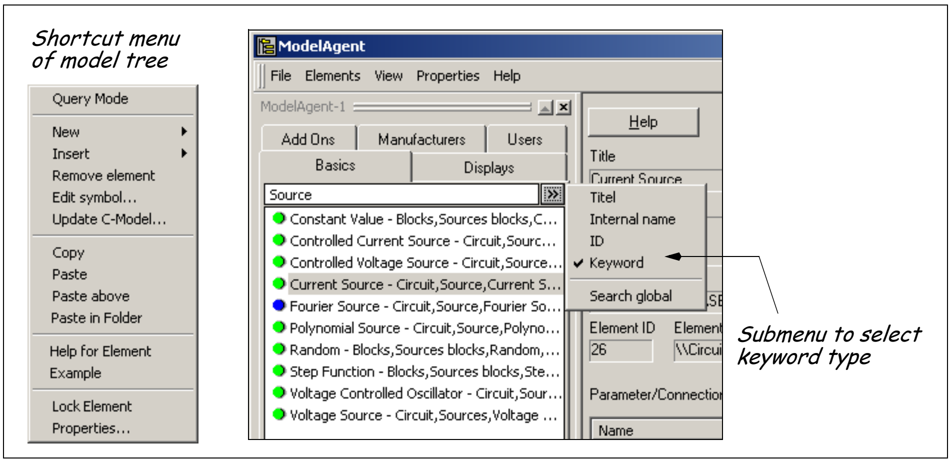
When you create a model, the Model Agent assigns a symbol that is used by Schematic to display the model wiring. You can change this symbol in the Symbol Editor by selecting the menu ELEMENTS>EDIT SYMBOL and making the desired change. See also 16 “Symbol Editor” on page 499.

Searching for Models

A search for models in the installed libraries can be time-consuming. The Search feature minimizes the complexity of the search and allows quick access to all models.

The Search feature is available only in Query mode. You can change modes via the shortcut menu or the VIEW>QUERY MODE and VIEW>TREE MODE menu commands.

In the search string field, the  button opens the submenu for the keywords used in the query mode.



Searches are possible for the following items:

Title	Specification in the model tree.
Internal Name	The name of the model, for example CNL, EI
ID	Internal ID Number of the model.
Keyword	Keyword, which identifies a model; a model can have several keywords.

After you have entered a keyword in the input field, the display beneath is updated immediately. The input is case sensitive. You can only look for keywords that are defined for the models. Newly created language resources or user-defined models have no entries for keywords, in which case you can only look for titles and internal names.

The search can be carried out within the selected library or in all installed libraries (a global search).

Creating Your Own Models (Components)

You can insert your own model in both default and user-defined libraries. There are three ways to create a new model:

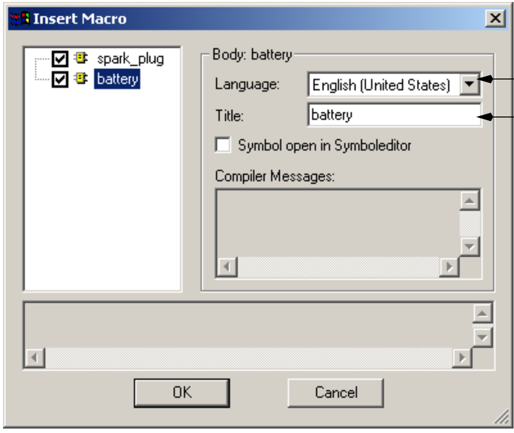
- From an existing SML model (.sml file)
- From an existing VHDL-AMS model (.vhd file)
- From an existing C model (.dll file)
- Directly in the Model Agent

Creating a Model from Existing .sml File

1. Create a .sml file with macro definition (at least one valid macro definition must be included). Subsheets in a simulation model are described as macro in the created .sml file (SIMULATION>DESCRIPTION and SML filter).
2. Open the Model Agent in the SSC Commander.
3. Select a library for the new model.
4. Choose ELEMENTS>INSERT>MACRO(S) FROM SML FILE in the Model Agent.

5 Select the .sml file with macro descriptions. The dialog displays all available macros definitions in the .sml file.

6 Select the box of a macro and define title and language. Repeat the steps if more than one macro is contained in the SML text. Click <OK> to import the macros.



Selecting the language and the macro title

Adapt the created symbol in the Symbol Editor. You can open them automatically if the Option **«Open Symbol Editor»** is checked. The model is completed and integrated in the selected library. It can be changed at any time as long as it is not encoded.

In the Schematic you can also create macro models and insert them in a library. See also “Creating Macros from Subsheet Models” on page 64.

Creating a Model from Existing .vhd File

1 Create a .vhd file with macro definition (at least one valid macro definition must be included). You can create VHDL-AMS files from models on a Schematic sheet using SIMULATION>DESCRIPTION and the .vhd filter.

2 Open the Model Agent in the SSC Commander.

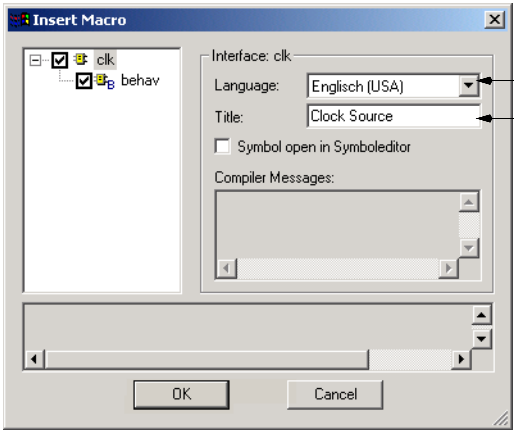
3 Select a library for the new model.

4 Choose ELEMENTS>INSERT>MACRO(S) FROM VHDL-AMS FILE in the Model Agent. This command is available only if the VHDL-AMS resource is already installed in the library (FILE>PROPERTIES).

5 Select the VHDL-AMS file with macro description. The dialog displays all available macro definitions in the VHDL-AMS file.

6 Define the title and language for the macro identified by the function, and click <Insert>.

7 Repeat the steps if more than one macro is contained in the source code. Click <Ignore> to skip a macro.



Selecting the language and the macro title

8 If desired, use the Symbol Editor to adapt the automatically created symbol. You can open the symbol automatically if the Option **«Open Symbol Editor»** is checked.

The model is completed and integrated in the selected library and can be changed at any time until it is not encoded.

In Schematic, you can also create macro models and insert them in a library. See also “Creating Macros from Subsheet Models” on page 64.

Creating a C Model from Existing .dll File (Optional Feature)

1 Create the .dll file with model definition (at least one valid model definition must be included).

2 Open the Model Agent in the SSC Commander.

3 Select a library for the new model.

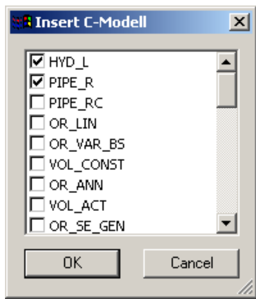
4 Choose ELEMENTS>NEW>C MODEL in the Model Agent

5 Select the .dll file with the model description.

6 Select the boxes for the corresponding C model.

7 Click <OK> to import the C models into the library.

The model is completed and integrated in the selected library. Adapt the created symbol in the Symbol Editor and add information to the parameters and nodes if you want. See also “Properties of Components” on page 493.



Select the boxes to import the C models

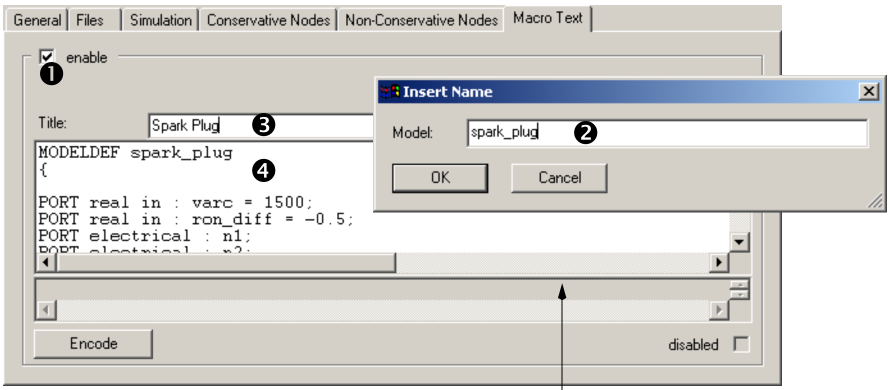
Creating Models in SML Text in the Model Agent

1 Select the library and destination folder.

2 Choose ELEMENTS>NEW>MACRO.

3 Click the **«General»** tab and enter a title (used for the model tree name).

4 Click the «**Macro Text**» tab and check the «**Enable**» box.
5 Enter a model definition name in the dialog box (used for the MODELDEF statement).
6 Enter a title (info line in the «**Library**» tab of components).
7 Enter the macro text into the «**Macro Text**» tab in SML syntax.



❶ *Check the «Enable» option*
❷ *Enter a model definition name*
❸ *Enter a macro title*
❹ *Enter the macro text in SML*

Display of syntax check messages

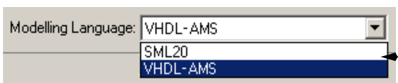
If the macro text is encoded with <Encode>, the text can no longer be changed. For encoded macros, the corresponding message is displayed in this input dialog.

8 Click <OK> to insert the macro into the library.

The macro is completed and integrated into the selected place in the model tree. Adapt the created symbol in the Symbol Editor and add information to the parameters and nodes if you want. See also “Properties of Components” on page 493.

Creating Models in VHDL-AMS in the Model Agent

1 Select the library and destination folder.
2 Choose ELEMENTS>NEW>MACRO.
3 Select VHDL-AMS as modelling language.



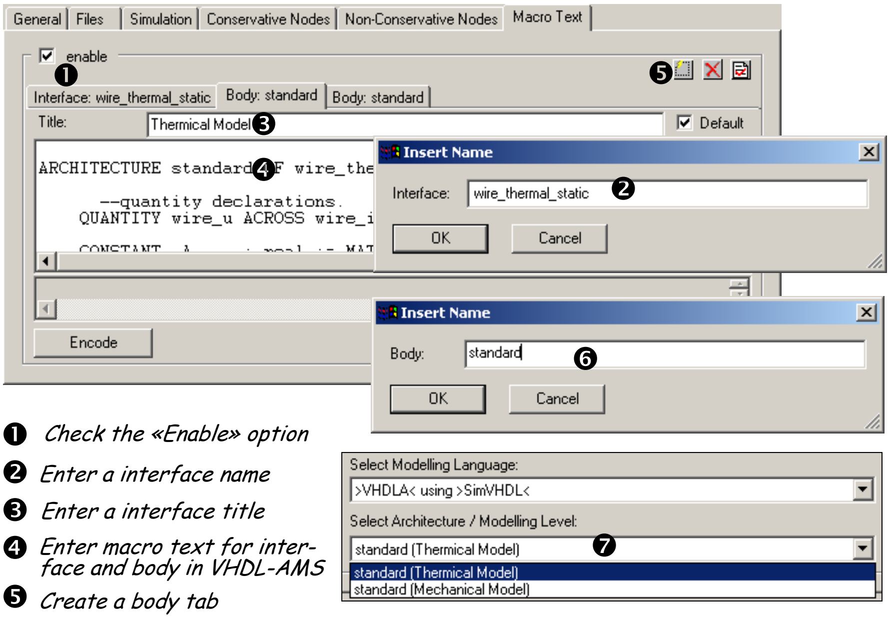
Selecting the modelling language

4 Click the «**General**» tab and enter a title (used for the model tree name).
5 Click the «**Macro Text**» tab and check the «**Enable**» box.
6 Enter a interface name and title.
7 Create a body tab with the symbols placed on the upper right side.

8 Enter a body name and title.

9 Enter the macro text into the **«Interface»** and **«Body»** tabs in VHDL-AMS syntax.

You can create **«Body»** tabs as many as you like. The different models can be chosen in the a dropped model on the Schematic sheet.



❶ *Check the «Enable» option*

❷ *Enter a interface name*

❸ *Enter a interface title*

❹ *Enter macro text for interface and body in VHDL-AMS*

❺ *Create a body tab*

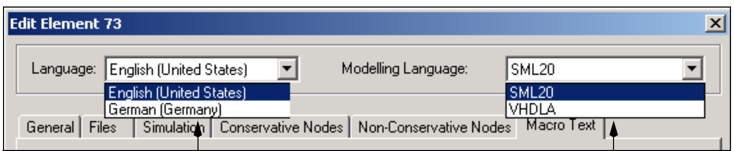
❻ *Enter a body name*

❼ *VHDL-AMS model with two body definitions in the Schematic sheet («Library» tab)*

Properties of Components

The ELEMENT>PROPERTIES command opens the property dialog of a selected component. You can also use the component shortcut menu to open the dialog.

All definitions filled in these dialogs are true only for the selected documentation language and modelling language.



Entries in the dialogs are applied to the selected documentation language (language for names and info lines)

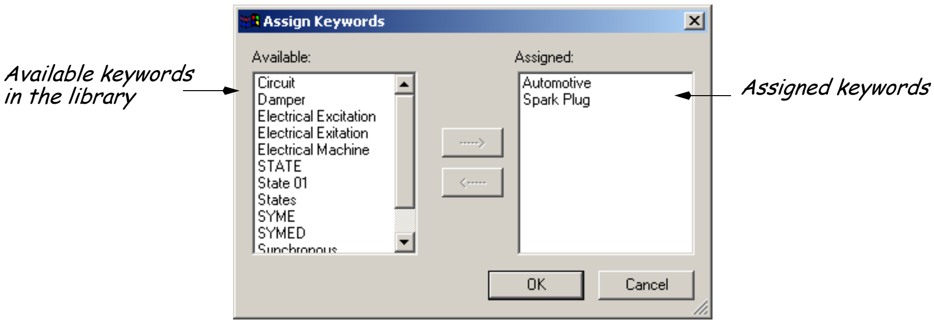
Entries in the dialogs are applied to the selected modelling language (language of the component description)

General Properties Tab

The **«General»** dialog contains general macro description. The Model Agent uses the title to display components (models) in the model tree. Keywords are used to find models in search mode. The entries in **«Designed by»** and **«Description»** have no function in other applications.

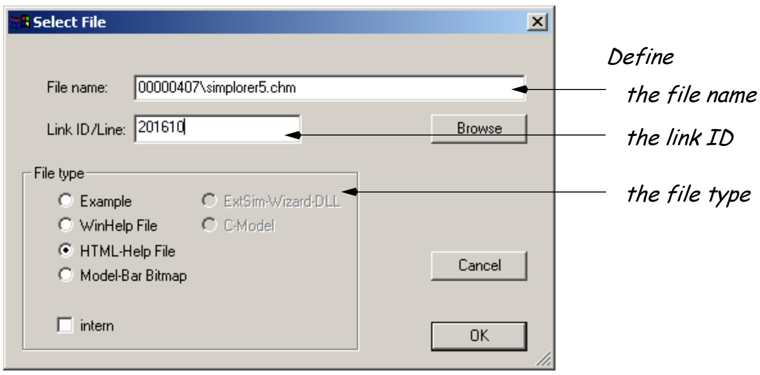
The name in **«Design by»** are always replaced with active user name when you close Model Agent.

With <Assign> you can display all keywords used in the library and assign them to the active component. If you need other keywords, click twice in the input field and enter the new entry.



Files Tab

The **«Files»** dialog contains links to all component files used, including language dependent files (list box above) and language independent files (list box below). The language independent files are used when no appropriate file is available for the active used language setting.



You can add, remove, modify, and test files using the buttons next to the corresponding list box. Define the type and name of the file and, if required, a component ID. The files have the following functions:

File	Function
User File	Defines a file which is connected with the model. All file formats can be included.

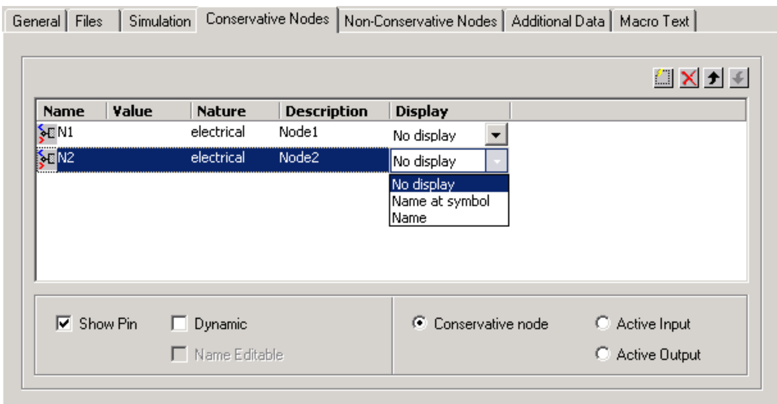
File	Function
Model Bar Bitmap	Defines a .bmp file 16x16 pixel used for the Model bar in the Schematic. See also "Modifying the Model Bar" on page 344.
WinHelp File	Defines a component help file in WinHelp format (.hlp file) and HTML Help format (.chm file). Only one file for each help format can be assigned. Both help files would be started when a link is provided. If you use relative paths, make sure that the SIMPLORER help directory is configured in the right way. See also "«Default Paths» tab" on page 21.
HTML Help File	
Example	Defines a file which is connected with the model. All file formats can be included. <Load Example> in the Working window opens all files assigned in this table. If you use relative paths, make sure that the SIMPLORER sample directory is configured in the right way. See also "«Default Paths» tab" on page 21.
C Model	Defines the .dll file used of the model. (Only for C models.)
☑ ☐ Intern	If the box is checked for a file assignment, the file is included completely in the model database, otherwise only the link is defined. Embedded files in the model library causes a larger file size, but you can exchange the library (.smd file) with all files.

Simulation Tab

The **«Simulation»** dialog contains the active macro call. You cannot make changes in this dialog. With <Edit> you can start the Symbol Editor to modify the component symbol. See also 16 "Symbol Editor" on page 499.

Conservative Nodes Tab

The **«Conservative Nodes»** dialog lists the connections (nodes) that are defined in the macro text, C model, or internal component. You can enter a common description and the display setting. To define a entry click the field in the respective column and enter the text. See also "Conservative and Non-Conservative Nodes" on page 44.

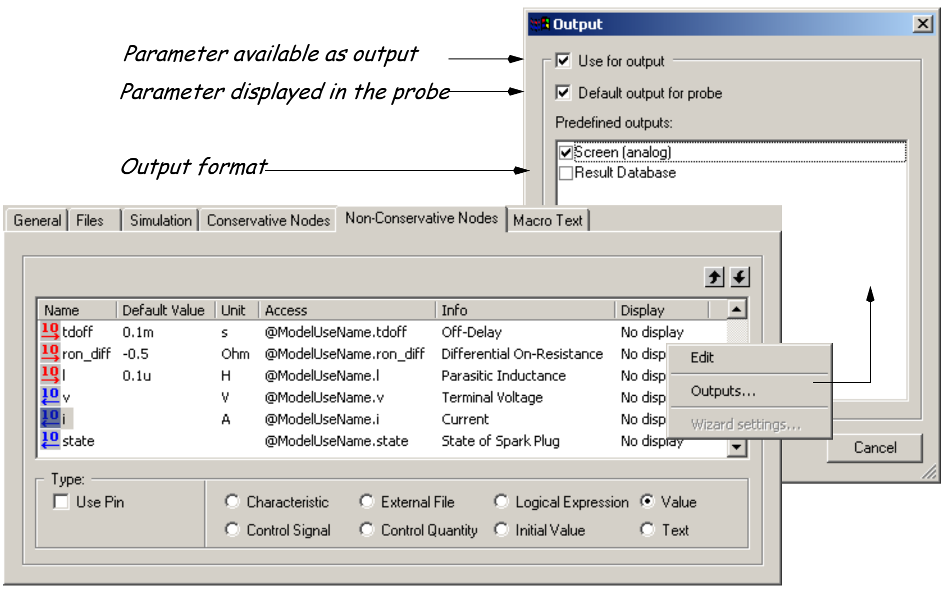


Non-Conservative Nodes Tab

The **«Non-Conservative Nodes»** dialog lists the parameters defined in the macro text,C model .dll file, or internal component. For a selected parameter, the figure used in the dialog can be changed by checking a box (Characteristic, External File, Logical Expression, Value, Control Signal, Control Quantity, Initial Value, Text) below the list field. The types have no influence on the parameter itself. The **«Use Pin»** option sets the input pin of a parameter as default when you drop the component on the sheet. See also “Displaying and Hiding Pins” on page 45.

You can also enter a parameter unit, a common description (**«Info»**), a default value, and a display setting. To define an entry, click the field in the respective column, and enter the text. The default value must adhere to the SML syntax (numerical value with or without unit suffix). See also 3.3.4 “Defining Component Properties” on page 48.

In case of C models, you can define units and descriptions in the model code itself. These are used for C model updates. Each parameter in the list has a shortcut menu, where you can define default output settings. Select one of the following output formats for a selected parameter: Screen analog or Result Database. If **«Default Output for Probe»** is checked, the parameter is displayed from the probe added to the component in the Schematic. You can define only one parameter for the probe. See also 12.2 “Display Elements” on page 369.



Macro Text Tab

The **«Macro Text»** dialog displays the SML or VHDL-AMS description of the component. You can edit and modify the macro text. The SML syntax of the text is checked when you click <OK>. Error messages are displayed below the input field. C models and encoded macro models have no text in this dialog box.

If the macro text is encoded using the <Encode> command in the dialog, the text can no longer be changed. In contrast to locked models, you cannot see the entries in the text field. For encoded macros, the corresponding message is displayed in this dialog.

15.4 Arranging Screen Layout and Using Help

Online help and version information are available with the Model Agent, as for all other SIMPLORER programs, from within the Help menu.

Arranging Windows

The Model Agent provides two independent model trees in a left and right window. You can display or hide the trees with the VIEW commands.

View Menu ▾

Command	Toolbar Symbol and Description
LEFT WINDOW	✓ Shows or hides the Left Window.
RIGHT WINDOW	✓ Shows or hides Right Window.
TREE MODE	✓ Starts the tree mode of the component representation.
QUERY MODE	✓ Starts the query mode of the component representation.

The Model Agent provides two independent model trees in a left and right window. You can display or hide the trees with the VIEW commands.

Properties Menu — Edit ▾

Dialog Tab	Description
«Symbol»	Defines if the symbol preview is displayed in the model tree and when Yes, the delay for display after the mouse point to the component title.
«Packages»	Show or hides VHDL-AMS package definitions in the model tree. See also ELEMENTS>NEW>PACKAGE.

Help and Version Information

As for all other SIMPLORER programs, online help and version information are available with the Model Agent from within the HELP menu.

Help Menu ▾

Command	Description
HELP CONTENTS	Go to online help.
ABOUT MODEL AGENT	Product and version information.

16 Symbol Editor

Every component in a model library has at least one symbol, which is displayed, when the component is dropped on the Schematic sheet. Component symbols are language dependent, that means, each language can provide its specify symbol.

With the Symbol Editor you can modify existing component symbols, such as the default symbol of a user-defined macro or the symbol of a SIMPLORER default component. Double-click the Symbol Editor  symbol or choose PROGRAMS>SYMBOL EDITOR to start the Symbol Editor in the SSC Commander.

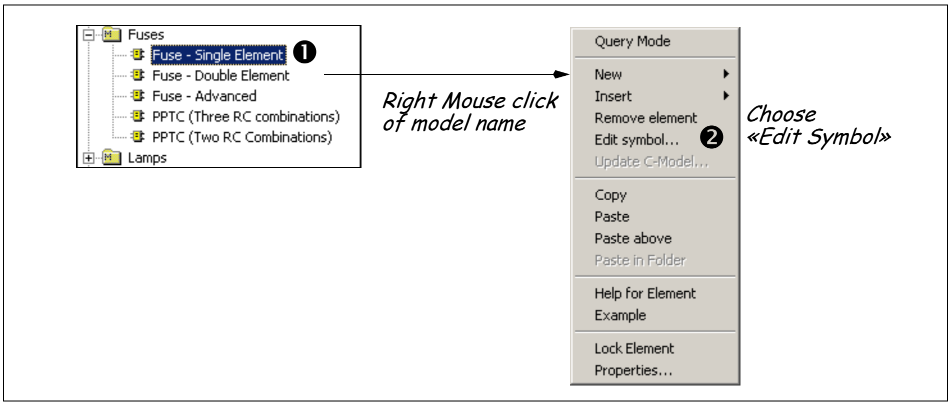
Along with the graphic properties you can also define optional features, such as switch areas and conditions for symbol changes during a simulation process.

This chapter contains information on:

- Starting the Symbol Editor
- Symbol Editor main window
- Parts of an element symbol
- Modifying symbols
- Editing pins
- Creating animated symbols
- Creating interactive symbols
- Modifying representations and displays

Starting the Symbol Editor

You can start the Symbol Editor in the Schematic or the Model Agent only for an **existing** model, such as an internal component or macro. The **«Edit Symbol…»** command on the shortcut menu of a component starts the Symbol Editor together with the selected model. For a selected model, the Symbol Editor can also be opened in the Model Agent with the menu command ELEMENTS>EDIT SYMBOL.

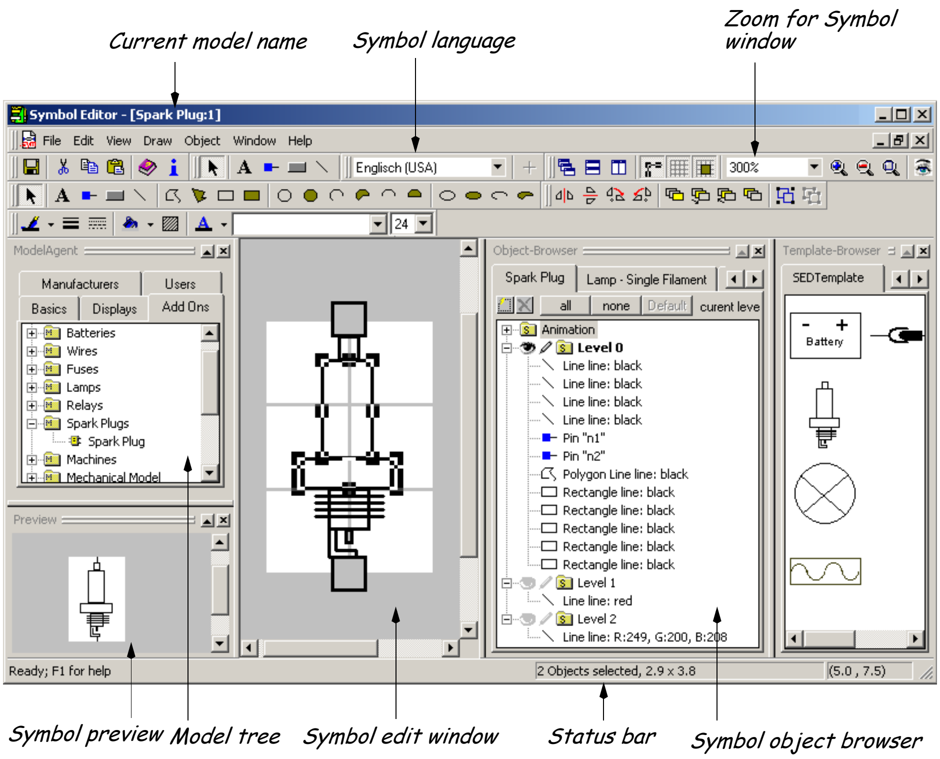


You can also start the Symbol Editor without a model in the SSC Commander. In this case the models can be loaded through drag-and-drop from the embedded model tree.

The Symbol Editor Main Window

The Symbol Editor main window consists of the following elements, which you can show and hide with the VIEW menu commands.

Symbol edit window	with active component symbols
Model tree	with installed libraries and components
Symbol object browser	with all parts of a symbol and their properties
Template browser	with predefined templates for symbol sheets
Symbol preview	with the representation displayed in the Schematic



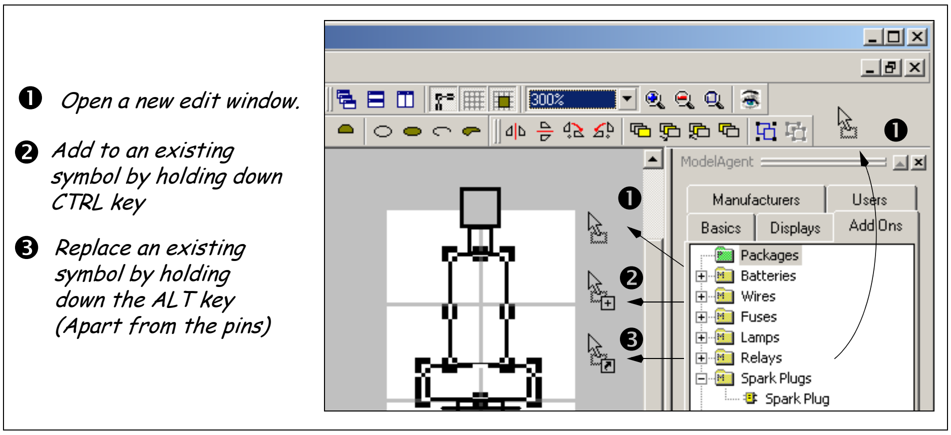
In the right pane the Object browser shows all the symbol's parts and its type. Each symbol part can be edited here using the shortcut commands «Delete», «Go to», and «Properties».

You can create several symbol levels, which you can show and hide with the buttons in the object browser. The choice of active level determines the level you can edit.

Symbol Management

You can load symbols when you start the Symbol Editor or within the Symbol Editor from the model tree. If you load symbols with drag-and-drop from the model tree, you have three options:

- Load the selected symbol into a **new** window (drop the element into the toolbar or the symbol edit window)
- Add the selected symbol to an **existing** symbol (drop the element into the symbol window by holding down the CTRL key).
- Replace the active symbol in the edit window with the selected symbol (drop the element into the symbol window by holding down the ALT key)



The FILE>SAVE menu command applies changes to symbol properties. The changes are effective if the symbol is dropped onto the Schematic sheet now. Components already existing in an opened simulation model in the Schematic can be updated with **«Synchronize»** on the components's shortcut menu.

File Menu

Command	Toolbar Symbol and Description	
CLOSE		Closes the active symbol with the prompt to save, if changes were made to the last saved version.
SAVE		Saves the active symbol, including all settings.
SAVE ALL		Saves all symbols open in the Symbol Editor.
SAVE AN GO BACK		Saves the active symbol and goes to the program started the Symbol Editor, for example the Schematic.
PROJECT		Switches to the SSC Commander window.
EXIT		Closes the Symbol Editor, prompting you to save if symbols are still open.

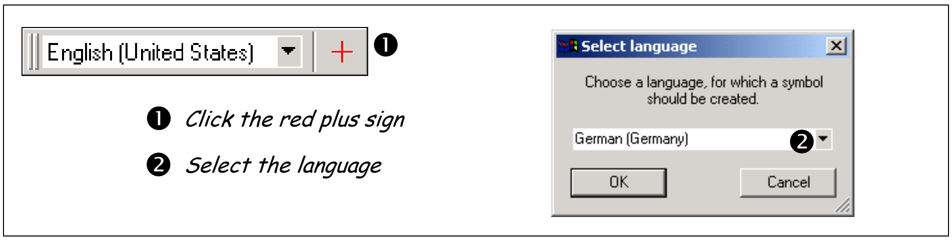
16.1 Modifying Symbol Sheets

All symbol properties —graphic elements, pins, interactions, animations— can be changed in the Symbol Editor. All settings saved in the symbol are visible if you drag the component on a Schematic sheet. You can modify symbols in two ways: Using the graphical interface of the Symbol Editor with all menu and toolbar commands or editing the properties within the EDIT>SYMBOL TEXT symbol text.

If you modify a symbol, changes are effective only for the language selected in the toolbar. You have to set the other language in the toolbar and repeat your inputs to adapt the available symbols to other languages.

The simplest way to copy a complete component symbol is to choose EDIT>SYMBOL TEXT, copy the text, change the language in the toolbar, and then insert the text using the EDIT>SYMBOL TEXT command for the other symbol. Then you will have an identical symbol and you only have to adapt the text strings which are contained in the symbol definition.

If the plus sign is highlighted red, there is at least one language without a symbol definition. Click the sign and select the language for which the standard symbol should be created. Provided that the plus sign is unavailable, all languages are available in the list and have an symbol definition. See also "SIMPLORER Language Concept" on page 483.



View of Symbol Features

Graphic Elements: Standard graphic elements, such as lines, circles, and text, which define the symbol layout.

Interactions: Change component properties (parameters) on the Schematic sheet.

Animations: Change the symbol layout during the simulation process, depending on the value of a system quantity.

Pins: Create connections to other components on the Schematic sheet.

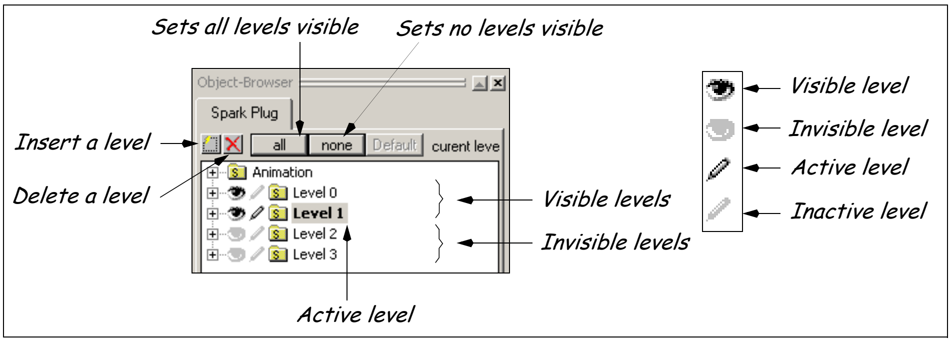
Graphic elements and interactions can be placed on different levels: 0 - 31; pins can be placed on level '0'. According to the definitions in the symbol, animations display different levels, separately or combined, on the Schematic sheet.

Activating and Displaying Levels

With the Object Browser buttons next to the level name, you can set the visibility and the edit mode of levels. Click the "eye" button to show or hide levels with different graphic elements. Click the "pen" button to activate the level. A level must be "active" to edit the elements contained in it. The active level is displayed immediately upon activation.

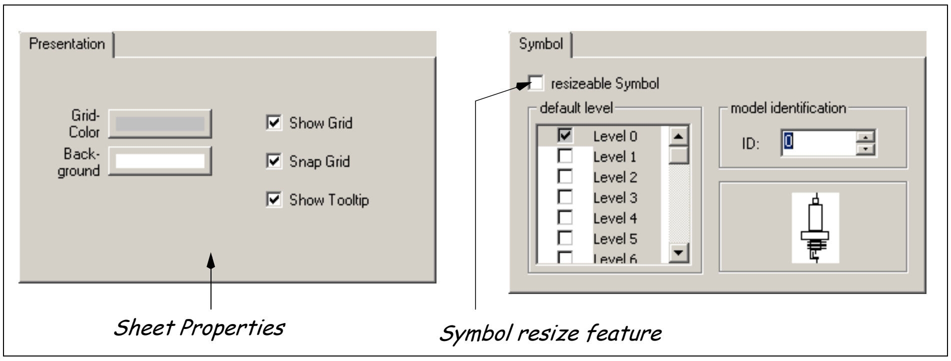
The lowest level is level '0'. Elements on the other levels are superimposed of elements in lower levels (if the lower levels are visible).

With the symbols on the upper side, levels can be created and deleted. The <All> and <None> buttons display or hide all levels of the active symbol.

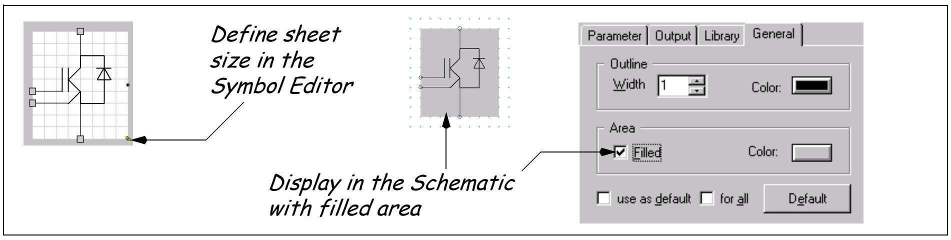


Defining Sheet and Symbol Properties

The settings in the VIEW>PROPERTIES menu are valid for the symbol sheet. The sheet properties and the sheet itself (size and form) can always be edited independent of the active level. The command is only available when the sheet is selected. In EDIT>PROPERTIES you can set the resize feature of a symbol on the sheet. If the option is checked, you can adapt the symbol size on the Schematic sheet.



You can adapt the sheet size by dragging on the blue sizing handles with the mouse pointer. The sheet should be large enough for all elements to be placed on it.



To place the elements exactly, you can display the grid and activate the Snap to grid function. All elements will then automatically be aligned with the grid. The grid size itself, however, cannot be changed.

Symbol Function	Toolbar Symbol and Description	
Display Grid		Displays the grid of the symbol sheet. The grid color is set in VIEW>PROPERTIES.
Snap to Grid		Aligns graphical elements to the grid.

16.2 Edit Functions

Through commands on the EDIT menu, symbols on the toolbar, or shortcut keys, all elements of the model sheet can be deleted or copied and inserted at a new location on the sheet. These functions always apply to selected elements.

Selecting Elements

When elements on the active level are selected, the object handles are visible (provided that the option VIEW>OBJECT HANDLES is selected). Several elements can be selected simultaneously by pressing the SHIFT key, or select all elements in an area by drawing a rectangle around them. In addition, there are commands on the Edit menu to select elements from one or more levels. Each selection can be cancelled by clicking anywhere on the sheet outside the selection.

Moving Elements

Place the mouse pointer in the yellow highlighted area and drag the selected components to a new position. If the **«Snap to grid»** box in EDIT>SETTINGS is checked, all elements are aligned with the grid.

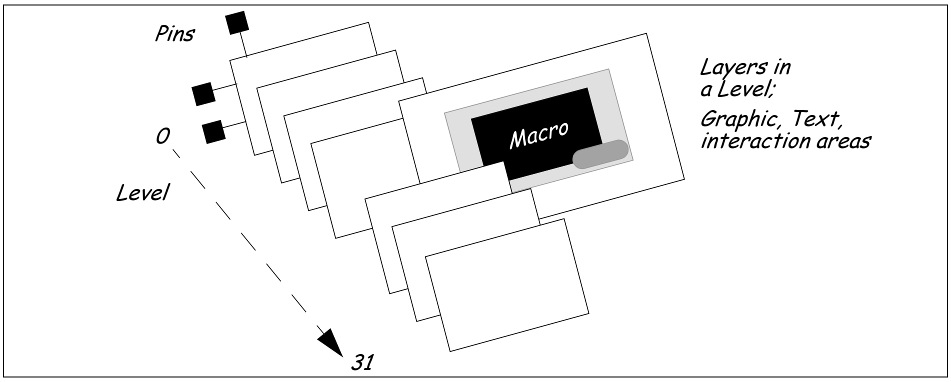
Edit Menu

Command	Toolbar Symbol and Description	
UNDO		Cancels or reverses the last input or action.
REDO		Restores an input or action.
CUT		Removes the selection and places it on the clipboard.
COPY		Copies the selected element to the clipboard.
PASTE		Pastes data from the clipboard on the active level or in the original level.
DELETE		Deletes selected elements.

Command	Toolbar Symbol and Description
SELECT	Selects the elements of the active level, the visible levels, or all levels.
SYMBOL TEXT	Shows the symbol properties in text notation. You can also edit all properties here.
PROPERTIES	Opens the property menu of the selected element.

Arranging Elements

Graphic elements and interactions can be placed on different symbol levels (0-31) and on different layers within a level. Within a level you can change the order of elements with the commands on the OBJECT menu. In the **«Level»** tab of an element property dialog, you can assign the level to an element. You can also choose EDIT>COPY and EDIT>PASTE to define a level number for elements. Pins are always placed on the '0' level and snapped to the grid.



Object Menu

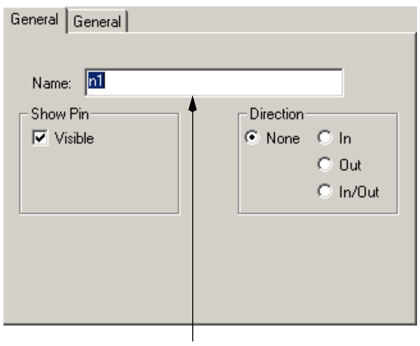
Command	Toolbar Symbol and Description
LINE COLOR	Changes the line color for the selected element.
FILL COLOR	Changes the fill color for the selected element
FLIP HORIZONTAL	Flips the selected element horizontally.
FLIP VERTICAL	Flips the selected element vertically.
ROTATE LEFT	Rotates the selected element 90° to the left.
ROTATE RIGHT	Rotates the selected element 90° to the right.
MOVE TO FRONT	Places the selected element into the foreground (within a level).
MOVE TO BACK	Places the selected element into the background (within a level).

Command	Toolbar Symbol and Description
MOVE FORWARD	Moves the selected element up one level.
MOVE BACK	Moves the selected element down one level.
GROUP	Groups selected elements.
UNGROUP	Ungroups elements.
ADD TO GROUP	Adds an element to a group.

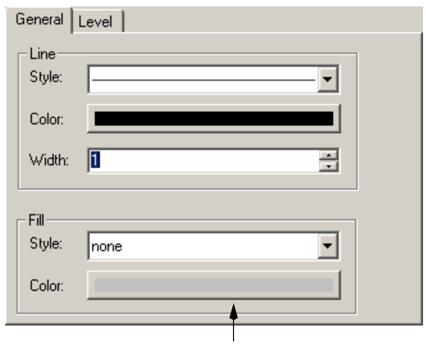
Defining Pins and Names

For a symbol selected from a library, all pins are normally visible. You can move, edit, and rename the pins later. Double-click a pin to open the property dialog. There you can define the name (displayed as a tooltip in the Schematic), and the line and fill color. If you insert an invalid pin (DRAW>PIN), the property dialog has *Invalid Pin* as name. The Symbol Editor always places a new pin at the sheet margin. Pins are always adjusted to the grid because of the working method of the Schematic wiring tool. The connection of the pin can be adapted to the symbol elements.

You can define several direction attributes for pins. **«None»** means no defined direction and is used for conservative nodes. **«In»**, **«Out»**, and **«In/Out»** are used for non-conservative nodes. On a Schematic sheet a small arrow indicates the direction.



Displayed tooltip in the Schematic



Properties of pins

Creating Graphic Elements

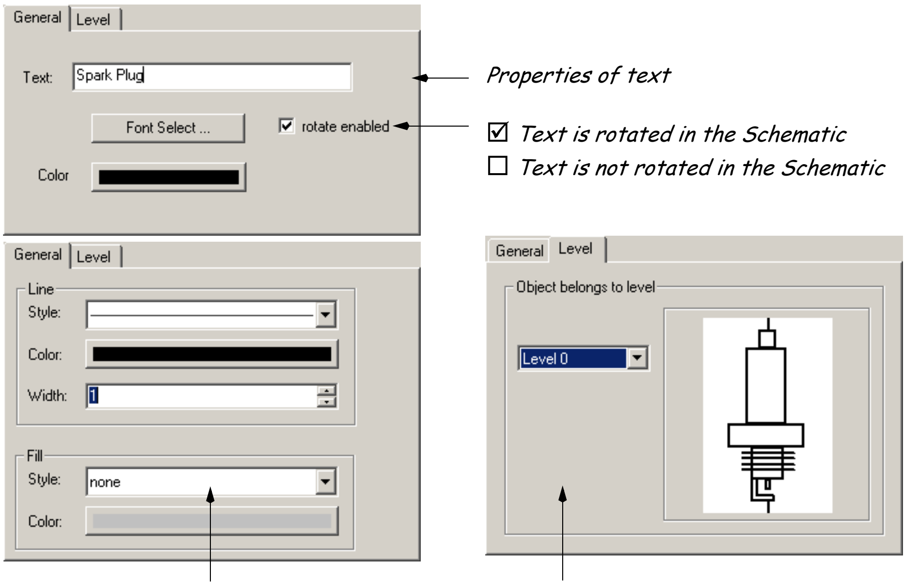
You can create several graphic elements (circle, rectangle, line, ...) in the Draw mode, which is started with the DRAW menu commands or the symbols on the Draw toolbar. Once in Draw mode, the mouse pointer changes into crosshairs. Lines, rectangles, ellipses, or polygons can be created on the sheet, according to the shape selected. If you select the polyline, set each corner with a mouse click.

The graphic elements have the same properties (color, line width, font) as in the PROPERTIES>EDIT dialog defined for the sheet. Double-click an element to open the property dialog, which contains the element settings for line width, fill, color, font, and level. Graphic elements are always inserted on the active level.

Draw Menu

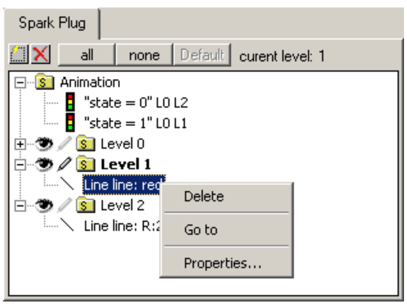
Command	Toolbar Symbol and Description
SELECT	Starts select mode (to alter or move shapes, lines, or text blocks).
LINE	Draws a line. Draw the line to the size desired by holding and then releasing the mouse button.
POLY	Draws filled or unfilled polylines and rectangles. You can change this property within the property dialog. To draw a polyline: Set the corners of the elements with the mouse. Close the mode with a double-click. The line are closed if the start and end points are placed at the same position. You can change the form if you move the corners with the mouse (the mouse pointer must be a cross arrow over the selected object). To draw a rectangle: Draw the rectangle to the size desired by holding the mouse button and the release the mouse button.
CIRCLE	Draws filled or unfilled circles, semicircles, and segments. You can change this property within the property dialog. Draw the element to the size desired by holding the mouse button and the release the mouse button. You can change the form of segments when you move the corners with the mouse (the mouse pointer must be a cross arrow over the red and green object handles of the selected element).
ELLIPSE	Draws filled or unfilled ellipses and elliptic segments. You can change this property within the property dialog. Draw the element to the size desired by holding the mouse button and the release the mouse button. You can change the form of elliptic segments when you move the corners with the mouse (the mouse pointer must be a cross arrow over the red and green object handles of the selected element).
TEXT	Creates a text element. Click in the sheet and enter the text. Double-click the text to open the property dialog. You can change font, color, and the behavior on the sheet (rotable or not).
PIN	Creates a new pin. Click in the sheet and the pin are placed at this position. You can insert pins only on level '0'. See also "Defining Pins and Names" on page 506.
INTERACTIVE PANEL	Creates an interactive panel. Draw the panel to the size desired by holding the mouse button and then release the mouse button. Double-click the panel to open the property dialog. See also "Creating Interactions" on page 510.

Command	Toolbar Symbol and Description
ANIMATION	 Creates a new animation feature for a symbol. If you start the command the property dialog is opened to define the control quantities and their limits. See also “Creating Animations” on page 509.



Properties of graphic elements *Level of a graphic element*

Object Browser

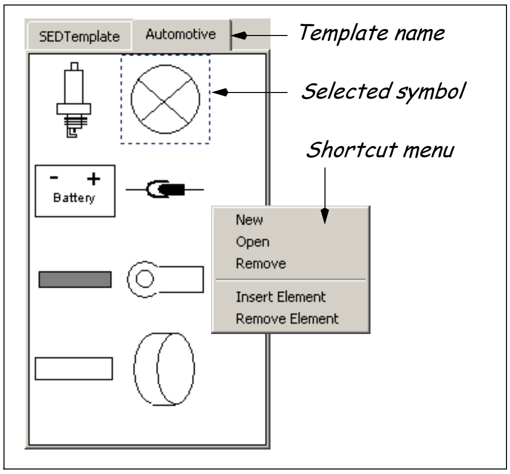


All elements contained in a symbol are displayed in the Object browser. Use the shortcut commands to select (**«Go to»**), edit (**«Properties»**), or delete (**«Delete»**) an element.Template Browser.

The template browser manages symbol templates, which contain several user-defined symbols. A symbol in a template can be dragged directly to a symbol sheet. The standard template is *SED template.tpl*.

Use the shortcut commands to create (**«New»**), open (**«Open»**), or close (**«Remove»**) a symbol template. The file extension of a symbol template should be .tpl. The extension, however, is not required for the function of the symbol template itself.

The shortcut command **«Insert Element»** adds the clipboard data as a new symbol to the template. You can add only symbols copied from a SIMPLORER symbol sheet.



The shortcut command **«Remove Element»** removes the selected symbol from the template. The preview of symbols in the template is standardized, so the real size is visible only on the symbol sheet.

16.3 Using Additional Features

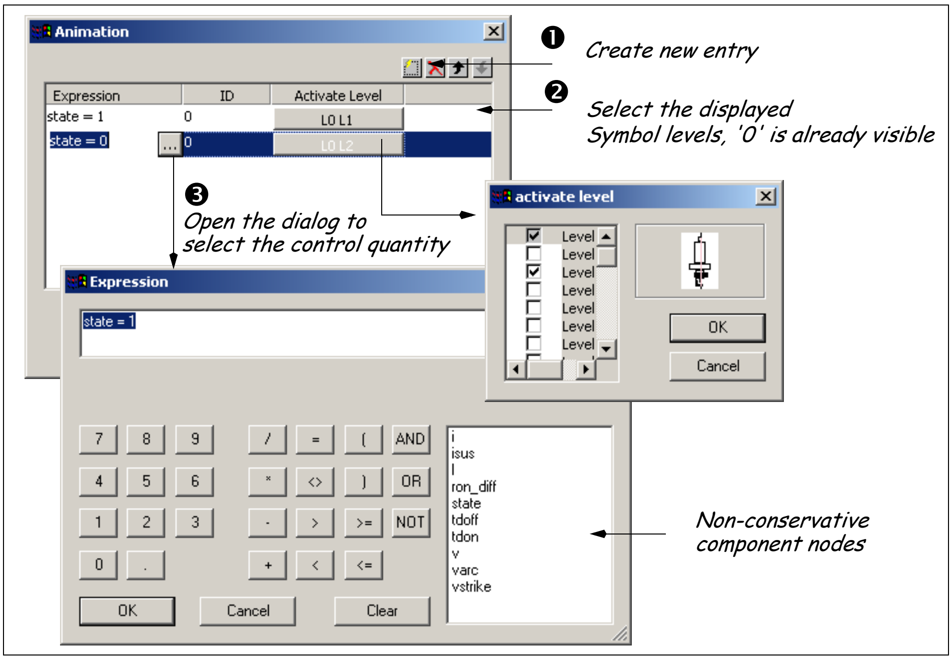
In addition to the graphic elements and pins, you can include animation and interactions in a symbol. Animations embedded in a symbol change the symbol layout during the simulation process, depending on the value of a system quantity. To display the states with animated symbols, select the **«Use Animated Symbols»** option in the SHEET>PROPERTIES **«System»** menu.

Using interactions you can change component properties (parameter values) of components placed on the Schematic sheet by clicking an interaction button.

Creating Animations

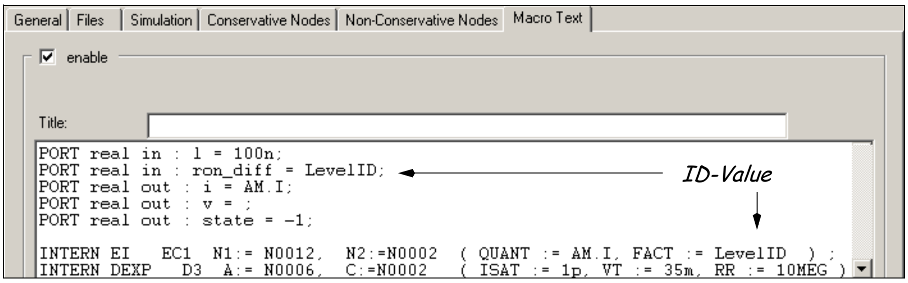
An Animated symbol changes shape during the simulation process, depending on parameter values (voltage, current, speed). For each control quantity used an upper and lower limit is defined, within certain symbol levels are displayed. The command DRAW>ANIMATION creates a new animation feature for a symbol. In the figure below the definition for an internal SIMPLORER component is explained.

To use animated symbols in the Schematic, select the option ☑ **«Animated Symbols»** in SHEET>PROPERTIES **«System»**.



Creating Interactions

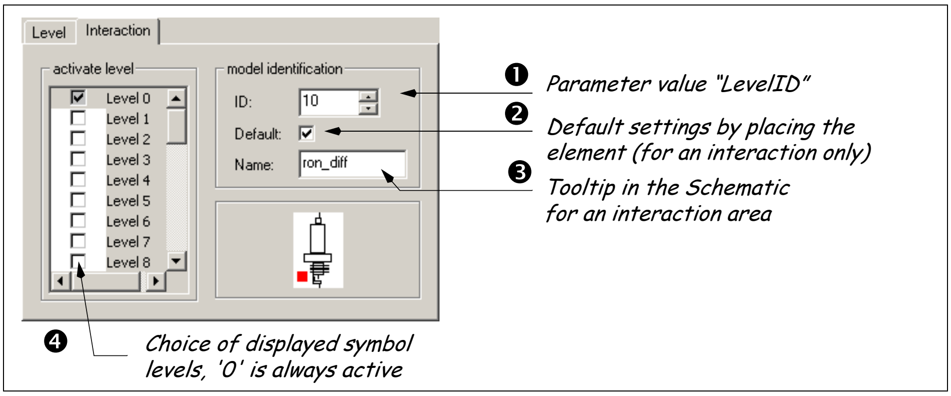
A symbol with interaction zones can be assigned parameter values before the simulation start by clicking on the interaction zone (e.g., setting the starting point in state graphs). In addition, the symbol's appearance can also be changed (e.g., a marked state has a blue point).



The interaction function is available only for macro models that have the entry "LevelID" in their parameter line. The LevelID parameter are assigned to the value, defined in the symbol. See also "Creating Your Own Models (Components)" on page 489.

Create an interaction area with the DRAW>INTERACTIVE PANEL command. Draw the panel to the size desired by holding and then releasing the mouse button. Double-click the panel to open the property dialog.

This rectangular area is visible in the Schematic only if the element is selected. When you click the area, the component receives the value defined in the **«Panel Properties»** dialog. The figure shows the definition of the ID value assigned to the "LevelID" parameter in the macro.



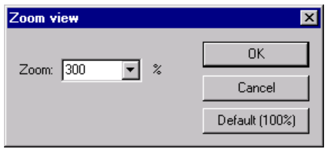
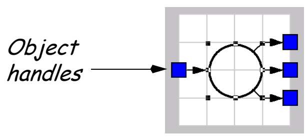
16.4 Arranging Screen Layout and Using Help

Online help and version information are available for the Symbol Editor, as for all other SIMPLORER programs, from within the HELP menu.

Arranging Symbols and Symbol Editor Elements

In the Symbol Editor, the commands to arrange the screen view of symbols and elements in the Symbol Editor window can be started from the VIEW menu or the toolbar symbols.

View Menu ▾

Command	Toolbar Symbols and Description
CHANGE ZOOM 	Opens the dialog for changing the symbol size used in the symbol window. 
ZOOM IN	Enlarges the active window by one step (10%).
ZOOM OUT	Reduces the active window by one step (10%).
COMPLETE SYMBOL	Enlarges the display of the symbol sheet to the maximum size.
UPDATE SHEET	Updates the screen display.
OBJECT HANDLES	Shows or hides the object handles of selected objects. 
PREVIEW	Shows the Schematic preview of a symbol. In the preview, the symbol cannot be modified.

Command	Toolbar Symbols and Description
OBJECT BROWSER	✓ Shows or hides the Object browser.
MODEL TREE	✓ Shows or hides the model tree.
TOOLBARS	✓ Shows or hides the Symbol Editor toolbars: STANDARD, DRAW TOOLS, DRAW LEVELS, and VIEW.
TEMPLATE BROWSER	✓ Shows or hides the template browser.
STATUS BAR	✓ Shows or hides the status bar with messages and key status.
PROPERTIES	Opens the property dialog to define settings of a symbol sheet.

Arranging Windows

In the Symbol Editor you can edit several symbols in different windows at the same time. The Window commands manage the arrangement and the appearance of windows on the screen.

Window Menu

Command	Toolbar Symbol and Description
CLOSE ALL	Closes all symbols open in the Symbol Editor, prompting you to save if changes were made since the last saved version.
CASCADE	Rearranges all open symbol windows in a cascade. All title bars are visible.
TILE HORIZONTAL	Rearranges all open windows side by side.
TILE VERTICAL	Rearranges all open windows one on top of the other.
ARRANGE ICONS	Moves all minimized windows to the lower part of Edit window and arranges them.
«LIST OF WINDOWS»	Lists all open windows and their corresponding symbol names.

Help and Version Information

Online help is available for all SIMPLORER programs, from the HELP menu or by clicking the symbol on the toolbar.

Help Menu

Menu	Toolbar Symbols and Description
HELP TOPICS	Go to the online help topics.
ABOUT SED	Product and version information.

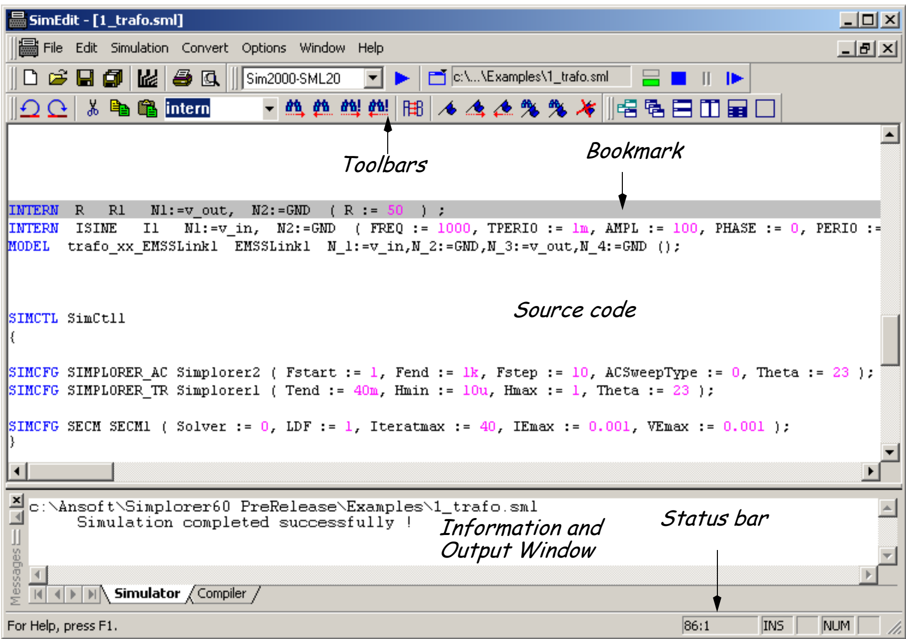
17 Text Editor

The Text Editor is an easy-to-use tool when you are working with the SIMPLORER modeling features. The input conventions are the same as those of typical text editors. The editor also contains special functions to integrate it into the SIMPLORER environment. Double-click the editor symbol or click PROGRAMS>EDITOR to start the text editor in the SSC Commander. If a project file is selected in the file list, a double-click starts the text editor with this file. If **{New File}** is selected, a double-click starts the Text Editor with either a new empty file or a new file from a template.

This chapter contains information on:

- Text Editor main window
- File management
- Spice Converter
- Text edit functions
- Using bookmarks
- Simulating models
- Defining settings

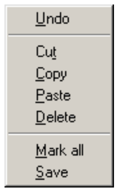
17.1 The Text Editor Main Window



By dragging the window sliders, you can adjust the size of the Input and Information windows on the screen.

The Information Window

When the Information window is visible, warnings and error messages from the compiler and the simulator are displayed. These messages help you find errors in the model description. In the Information window, a shortcut menu can be opened with a right mouse click in the window. The messages can be copied into the clipboard and inserted into any editor.



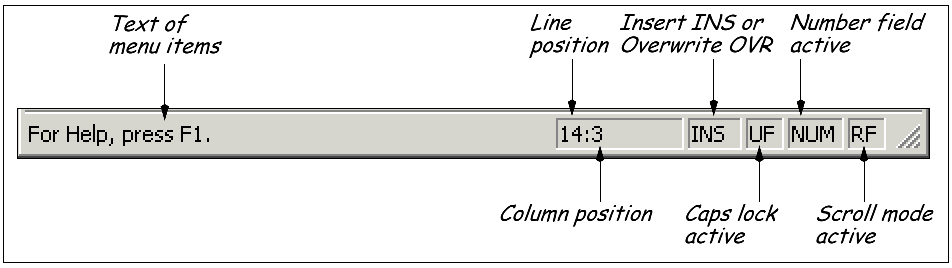
Changes made to the text in the Information window have no influence on the source code for the simulation.

The Simulator Queue

Usually the simulation model are computed immediately after the simulation call. If there are other simulations still running, the new task is placed in the simulator queue. If you wish, you can change the sequence of tasks in the queue with the commands in the SSC Commander.

The Status Bar of the Text Editor

The status bar is located at the bottom of the Text Editor window. It displays the active key settings and the cursor position in the document. At the left side, brief information about the corresponding menu item is displayed.



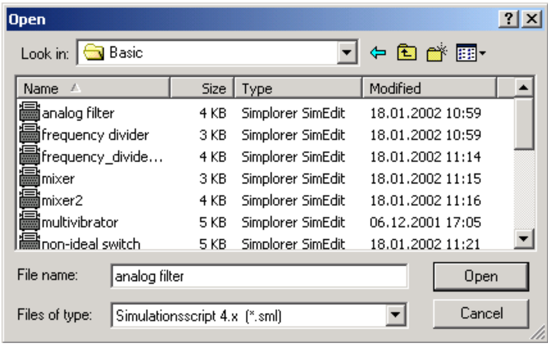
17.2 File Management in the SIMPLORER Text Editor

The commands to load, save, and print files can be opened with the FILE menu or symbols. The Text Editor is able to load all text or ASCII files and to insert the contents of the Windows clipboard from other applications. Files created or modified in the Text Editor can be included in the active project.

The file filters which are used in FILE>OPEN and FILE>SAVE are entered in the Text Editor through OPTIONS>SETTINGS **«File Options»**.

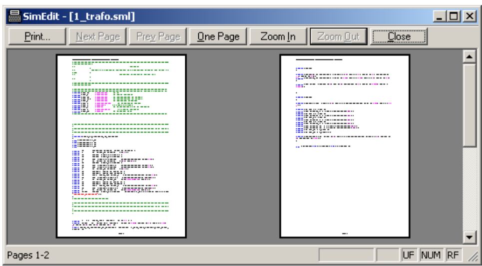
File Menu

Command	Toolbar Symbol and Description
NEW	Opens a new window to create a document.
OPEN	Opens an existing file; the displayed list can be modified with filters for file extensions.



Open File dialog box

Command	Toolbar Symbol and Description
CLOSE	Closes the active document with prompt to save, if any changes were made to the last saved version.
SAVE	Saves the active document. You will be prompted for a file name if the document is being saved for the first time.
SAVE AS	Like SAVE, except a file name is always required.
SAVE ALL	Saves all documents open in the Text Editor. If necessary, the file name must be entered.
PROJECT	Switches to the SSC Commander window.
PRINT	Prints the active document. Defines the print pages, the number of copies, and the print device.
PRINT PREVIEW	Displays the document in full-page view.



Print Preview Dialog in the Text Editor

Command	Toolbar Symbol and Description
PRINTER SETUP	Defines the print options: paper size, paper feed, and format (portrait or landscape).

Command	Toolbar Symbol and Description
«FILE LIST»	Lists the last files opened. The number of entries depends on the value in OPTIONS>OPTIONS>FILE OPTIONS **«Number of recent files»**.
EXIT	Closes the Text Editor with prompt to save if documents are still open.

17.3 SPICE Converter

The SPICE converter, a semiautomatic conversion tool, is part of the SPICE-Pack package. It is implemented as an extension of the SIMPLORER Text Editor. The SPICE converter converts SPICE netlists into SIMPLORER SML descriptions to include SPIC3F5 semiconductor models into SIMPLORER. SPICE subcircuits become SIMPLORER macros. The solver library is implemented as a .dll file.

Converted SPICE netlists with semiconductor components can only be simulated with the optional SIMPLORER SPICE-Pack. For more information please contact our sales team at sales@simplorer.com.

Converting SPICE Netlists

After loading the SPICE netlist you can start the conversion with CONVERT>SPICE->SML 2.0. When the conversion process is finished a second window is opened containing the new SIMPLORER SML netlist. At the same time a file is created using the same file name as the .cir file, but the file extension is changed to .sml. Since almost each SPICE-based simulator has its own dialect (extensions and differences to the SPICE 3F5 standard) it is impossible to support all existing statements of the respective software packages. Therefore, some statements that cannot be translated are placed in the new .sml files as a comment. A message is displayed during the conversion process. In these cases you have to replace the statement by a SIMPLORER model providing an equivalent behavior.

Unlike semiconductor models in SIMPLORER that are usually implemented using the C/C++ programming interface, SPICE semiconductor models are often subcircuits with a variety of basic components. Each of these subcircuits must be translated into a SIMPLORER SML netlist. The components of the SPICE-Pack extension correspond to the properties and parameters of the original SPICE components. Basic elements such as resistors, inductors, controlled sources, and others can be replaced by SIMPLORER basic components without any change. All semiconductor components are provided by the SPICE-Pack extension.

Creating a Model from a SPICE Netlist

1 Load the SPICE netlist in the Text Editor.
2 Choose CONVERT SPICE>SML 2.0 to convert the text.
3 Check the created SML description. If comments for non-translated statements were created replace the statement by a SIMPLORER model providing an equivalent behavior.
4 Open the Model Agent in the SSC Commander.
5 Select a library for the new models.

6 Choose ELEMENTS>NEW/MACRO(S) FROM SML FILE in the Model Agent and open the .sml file with the converted SPICE models.

7 Click <OK> to accept the suggested symbol.

Repeat step 6 and 7 if more than one macro model is contained in the SML description. The model is completed and integrated in the selected library. It can be changed as long as it is not encoded. See also "Creating Your Own Models (Components)" on page 489.

Convert Menu ▾

Command	Description
SML 1.5 ->SML 2.0	Converts a version 4.x SML description to 5.0/6.0.
SPICE -> SML 2.0	Converts SPICE netlists into SIMPLORER SML 2.0 descriptions.

Available Components

The capability of the SPICE converter has been tested with models for diodes, MOSFETs, and IGBTs from International Rectifier. The SPICE netlists are available at http://www.irf.com/qlmodel.html. The models were completely translated and verified in SIMPLORER. The SIMPLORER IR libraries are available for download. See also "Converter List and Non-Supported Components" in the Online Help.

17.4 Modifying and Editing Text

The SML source code consists of comments, preprocessor directives, and sections. The sections start with a keyword and contain text lines. The Text Editor is able to recognize and colorize special SML keywords in the source code. The automatic text highlighting is active only when you have saved the source code as an .sml file. You cannot change the colors in the Text Editor.

Using Templates

Templates which already contain keywords and comments in which to enter the SML text can be helpful in creating SML code. To automatically load a template in a new window when you start the Text Editor, choose OPTIONS>PROGRAM DIRECTORIES in the SSC Commander and enter path and name for the Editor's template file.

Drag-and-Drop from the Model Agent

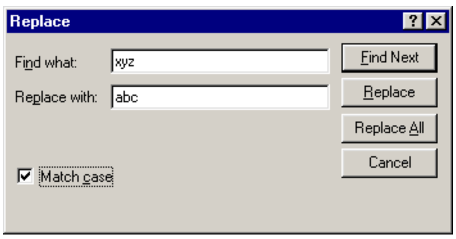
Another option for inserting SML source code quickly and accurately is to use components from the model tree in the Model Agent and in the Schematic. If the model tree and the input window of the Text Editor are arranged side-by-side on the screen, you can drag-and-drop components of the model tree into the Text Editor window. The SML notation for the component or the macro call are inserted at the position of the mouse pointer. If you have checked the option «Add comment» in OPTIONS>OPTIONS>GENERAL OPTIONS, the respective comments from the model are also inserted.

Editing the Text

In the SIMPLORER Text Editor, many functions are available to modify text. These functions can be started with the EDIT menu or symbols in the toolbar.

Edit Menu

Command	Toolbar Symbol and Description
UNDO	Reverses or cancels the most recent actions. The maximum number of actions you can undo is defined in OPTIONS>OPTIONS>GENERAL OPTIONS.
REDO	Restores the most recent action of UNDO.
CUT	Cuts the selected text and places it in the clipboard.
COPY	Places the selected text in the clipboard.
PASTE	Pastes the text in the clipboard into the active document at the cursor position.
SELECT ALL	Selects the entire text of the active document.
DELETE ALL BOOKMARKS	Deletes all existing bookmarks in the active window.
FIND	Searches for the entered string with the options: • Search direction (up or down) • Match case
REPLACE	Similar to Find except that the search string can be replaced by another string.



Search and Replace Dialog

GO TO	The cursor goes to the entered page number.
OEM->ANSI	Changes the text to ANSI characters.
ANSI->OEM	Changes the text to OEM characters.

The commands COPY and CUT always apply to selected text. These commands are available only if text is selected in the document.

Key and Mouse Functions

Selecting Text

There are two ways to select text:

- Draw a rectangle around the text while holding the left mouse button.
- Press the SHIFT key and select an area with the arrow keys.

The selected area are highlighted on the screen.

Text Edit Column Wise

If the CTRL key is pressed **before** selecting text, then the text are selected in columns. To insert the text in columns at the cursor position, the key combination CTRL+SHIFT+INS must be used.

Moving Text

Selected text can be dragged with the mouse to a new position. In a valid insert field, the mouse pointer appears as a rectangle. If the CTRL key is pressed during the movement, the text is duplicated at the new position.

17.5 Using Bookmarks

Bookmarks are colored lines which are inserted at certain positions in the text. Once created, these bookmarks allow quick access to text sections. In contrast to the FIND command, which searches for any occurrence of a text string, the bookmark is connected with a certain position. Especially in extensive source codes, orientation is easier and searching the text is faster with bookmarks. Bookmarks can be inserted and removed with the toolbar symbols. _

The search string can be entered in EDIT>FIND or directly in the toolbar. If a search string in the text is identical, the criteria to insert a bookmark is valid.

Symbol Function	Toolbar Symbol and Description
Toggle bookmark	Inserts a bookmark at the active cursor position, or turns off an existing bookmark. (Define the color in OPTIONS>BOOKMARK COLOR.)
Bookmark forward	Moves the cursor to the next bookmark in the text.
Bookmark backward	Moves the cursor to the previous bookmark in the text.
Set bookmark	Inserts a bookmark at all positions where the active search string occurs.
Set bookmark case	Similar to Set bookmark, but case sensitive.
Delete bookmarks	Deletes all bookmarks in the text.

17.6 Starting a Simulation

When the simulator is opened from the Text Editor, the SML compiler translates the created model into legible code for the simulator and then start the simulation. With every simulator call the active file are saved. When saving for the first time, the file name must be entered.

During a running simulation, the file name of the model is visible in the simulation toolbar and the symbols to stop and break the simulation are available. See also 13 “Simulator” on page 397.

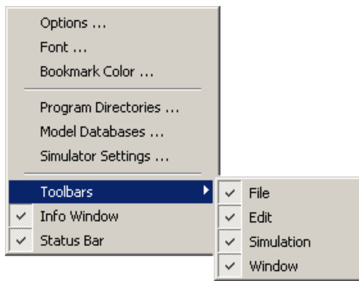
Simulation Menu ▾

Menu	Toolbar Symbol and Description
START	Starts a simulation with the active SML source file.
STOP	Final stop of a running simulation.
BREAK	Temporarily breaks or pauses a running simulation.
CONTINUE	Resumes a paused simulation.

17.7 Settings and Options

Users can adapt the program environment and the specific editor settings to their individual preferences through the OPTIONS menu. The settings are valid until they are changed by the same user (identified by user name when the program starts).

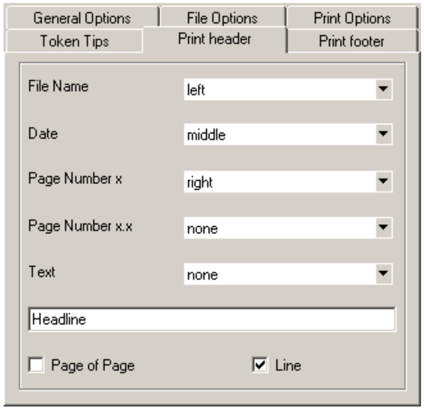
In addition, you can display or hide the toolbars, the Information window, and the status bar.



Options Menu ▾

Command/Tab	Description
SETTINGS	
«General options»	• Saves the window position. • Inserts spaces at the beginning of the line and also at the beginning of the next line. • Deletes spaces at the end of line when saving. • Defines the number of undo actions. • Inserts model comments from the Model Agent.

Command/Tab	Description
«File options»	• Defines the number of files which can be started directly in the FILE menu. • Creates additional file filters for selection in the FILE>OPEN and FILE>SAVE AS menus.
«Header/Footer line»	• Contents, position, and layout of header and footer lines defined by selecting options.
«Print options»	• Prints line numbers. • Defines the left and right margins of the print page.
«Token Tips»	• Switches the Tips display on or off. • Layout (delay, border, color, font).
FONT	Selects the font and font size for the display and the print page.
BOOKMARK COLOR	Changes the color of the inserted bookmarks.
PROGRAM DIRECTORIES MODEL DATABASES SIMULATOR SETTINGS	These menu items are identical with the menu OPTIONS of the SSC Commander
TOOLBARS	Switches these toolbars on or off: FILE, EDIT, SIMULATION, and WINDOW.
INFO WINDOW	Switches the Information Window on or off.
STATUS BAR	Switches the Status bar on or off.



17.8 Arranging Screen Layout and Using Help

The Text Editor provides settings for arranging Windows. You can define the settings within the WINDOW of the Text Editor. See also 17.1 “The Text Editor Main Window” on page 513.

Arranging Windows

In the Text Editor, several files can be edited at the same time in different windows. The arrangement of the windows on the screen is managed with the window commands, accessible from the toolbar symbols or from the WINDOW menu.

Window Menu

Command	Toolbar Symbol and Description
NEW WINDOW	Creates a duplicate of the active window.
OVERLAPPING	Rearranges all open windows so that they overlap (cascade). All title bars are visible.
TILE HORIZONTAL	Rearranges all open windows one on top of the other.
TILE VERTICAL	Rearranges all open windows side by side.
ARRANGE ICONS	Moves all minimized windows to the lower part of the Edit window and arranges them.
CLOSE ALL	Closes all open documents.
«LIST OF OPENED FILES»	Lists the open windows and their corresponding file names.

Help and Version Information

Online help is available for all SIMPLORER programs, from the HELP menu or by clicking the symbol on the toolbar.

Help Menu

Command	Description
HELP CONTENTS	Go to the table of contents in the online help.
ABOUT SIM EDIT	Product and version information.

Appendix

A.1 Glossary

Term	Description
AC simulation	harmonic analysis of a model.
AC simulator	SIMPLORER simulator for AC analysis.
Action type	defines how an action in a state is processed.
.afa file	Analytical Frequency Analysis file; containing data of an analytical frequency analysis.
Analytic frequency step response	SIMPLORER module to compute frequency step response information for a given transfer function.
Animated Symbol	symbol for a component or a macro that changes to reflect changes in values assigned to it. The symbol can be modified by the user with an interaction button or by a system value during the simulation. Animated symbols can be generated using the Symbol Editor.
.aws file	Simulator file; containing initial values of capacitors and inductors.
Backplane	common communication and control structure for all SIMPLORER simulators.
Basic version	SIMPLORER basic version without any optional modules.
Basics	SIMPLORER library containing the basic modeling elements for the SIMPLORER simulators.
Behavioral model	smallest model unit that cannot be subdivided further.
Best representation	function for scaling all output channels of a graph window to maximum size.
Bezier	Bezier curves are used in computer graphics to produce curves which appear reasonably smooth at all scales. The curves are constructed as a sequence of cubic segments, rather than linear ones. Bezier curves use a construction in which the interpolating polynomials depend on certain control points. The mathematics of these curves is classical, but it was a French automobile engineer Pierre Bezier who introduced their use in computer graphics.
Block	linear or nonlinear transfer function, basic element of block diagrams.
Block diagram	combination of blocks to describe the dynamic behavior of systems.
Block Diagram Module (BDM)	SIMPLORER sub simulator to analyze simulation models described using block diagrams. Uses Euler formula and distributed integration algorithms.
Bookmark	can be used in the SIMPLORER editor to mark a position and find it again.
.brs file	External Schematic file to parametrize a model sheet.
Call back function (UBF)	function type for the definition of C/C++ simulation models.
Call sub function (UCF)	function type for the definition of C/C++ simulation models.
Characteristic	function or data set to describe a nonlinear characteristic of a SIMPLORER component, block or other model.
Characteristic values	analysis method in the DAY Post Processor; provides a list of frequently used characteristic values of a system quantity.
C-Interface	C/C++ programming interface for the integration of user defined nonlinear models or components into the simulation model.
Color scheme	predefined or user defined color settings for screen outputs or printing.
Compiler	program for the translation of the SML description of a simulation model into a simulator specific format.
Computation sequence	sequence in which blocks in a simulation model are computed.
Connection rule	rule determining which element can be connected to another under certain conditions.
Conservative node	connection of two or more circuit component terminals.
Coordinate system	reference system for the display of numerical values; can be linear or logarithmic.

Term	Description
Crossing over	exchange of genetic material between chromosomes in a genetic algorithm.
Cursor	positioning element to determine the value of a quantity in a coordinate system.
Data cache	saves the data of the last simulation run.
Data channel	simulation data for a specific quantity stored in a file or transferred to an active element.
Data filter	function to select data by user defined criteria.
Data format	defines how data are stored or exchanged between programs.
Data set	simulation data of a simulation run at a given time step.
.day file	DAY Post Processor file; containing data of a data analysis.
DAY Optim	SIMPLORER module for the evaluation of optimization results.
DAY Postprocessor	SIMPLORER module for the post processing of simulation and measurement data.
DC simulation	DC operating point analysis of a model.
DC simulator	SIMPLORER simulator for DC analysis.
Differentiation	analysis method in the DAY Post Processor.
DISPLAY	SIMPLORER library containing display elements to display simulation data inside a schematic.
Display Element	element for graphic or numeric online visualization of simulation data on sheet.
DLL	dynamic link library; contains program components or models; loaded automatically upon request.
Dongle	software protection device connected to the printer port of the computer.
DSDE	Dynamic Simulation Data Exchange interface; used to integrate external simulators on a client-server-level.
Electric circuit	combination of electric components, connected by ideal wires.
Electric Circuit Module (ECM)	SIMPLORER sub simulator to analyze simulation models described using electric circuits. Uses Euler or Trapezoidal formula and modified nodal approach.
Element	element of the graphical representation of the simulation model.
Entity	VHDL term to describe the interface of a behavioral or structural model.
Euler formula	numerical algorithm used inside SIMPLORER.
Evaluation function	function evaluating the quality of a solution compared to the defined optimum.
.exp file	file containing the settings of an experiment.
Experiment tool	SIMPLORER module for the definition and management of experiments performed using a model.
Experiment wizard	special wizard for an easy definition of experiments.
Expert mode	extended mode for the definition of an experiment; allows fine tuning on parameters and methods.
Export filter	special program for the export of simulation data into other applications.
Extern View	special sheet independent oscilloscope for the display of simulation data.
FFT	Fast Fourier Transformation.
.fil file	DAY Optim file; containing data of a filter applied to the data file.
File output	saves a system quantity online during the simulation in a file.
Fitness function	function describing the fitness of an individual (parameter constellation) in a genetic algorithm.
Formula Module (FML)	SIMPLORER's integrated expression evaluator.
Frequency step response analysis	special mode in the Experiment tool, determining the frequency behavior of a simulation model.
Genetic Algorithm	optimization method of the experiment tool with automatic parameter variation and target function determination.
Grid lines	grid lines in a coordinate system.

Term	Description
HMAX	simulation parameter, maximum step size for the integration algorithm.
HMIN	simulation parameter, minimum step size for the integration algorithm.
ID	numeric value for the distinction of sheet elements.
Information window	display of warnings, errors, program status for the SIMPLORER modules.
Initial condition	defines the initial value for energy storing electrical components.
Initial state	a state that is active at the beginning of the simulation.
INT	Simulation parameter, defines the integration algorithm used for the simulation of electric circuits.
Integration	analysis method in the DAY Post Processor.
Interactivity pad	part of an animated symbol, used to change the behavior and/or shape of a symbol (model component) by the user.
Iteration	part of the integration process for the solution of nonlinear problems.
Jacobian Matrix	coefficient matrix for the numerical integration algorithm.
Key word	can be used to do a search in a model database.
.krn file	Simulator file; containing simulation status information; can be used as a starting point for the next simulation.
Language concept	Settings to define the use of language for program (menus and dialogs) and libraries.
Library	database containing a set of SIMPLORER basic elements and/or macro models.
Local discretization error (LDF)	simulation parameter, determines the accuracy of the computation according to the dynamics of the electric circuits.
.log file	Experiment tool file; containing the protocol of an experiment.
Macro	contains one or more elements of a model description that can be used as single elements.
Main skeleton (.skl)	special file containing basic descriptions for the generation of the model description file. Must not be modified.
Maximum current error (IFMAX)	simulation parameter; determines the accuracy of the right side computation of the differential equation system for current lines.
Maximum voltage error (UFMAX)	simulation parameter; determines the accuracy of the right side computation of the differential equation system for voltage lines.
.mda/.mdk file	SIMPLORER data file of simulation results and characteristics (former format).
.mdx file	SIMPLORER data file of simulation results and characteristics.
Model	representation of the behavior of a real system using mathematical description methods.
Model tree	hierarchical list box containing the available elements for the graphical modeling.
ModelAgent	SIMPLORER module for the model library management.
Module	SIMPLORER sub simulator or program.
Monitor	special window for the display of simulation status and progress.
Monte Carlo Analysis	optimization method of the experiment tool with automatic parameter generation and characteristic value determination.
.mtx file	SIMPLORER data file of the DES model.
Multi simulation	analysis method of the experiment tool, batch mode computation of a simulation model with different, user defined parameter sets.
Mutation	part of the genetic algorithm; generates new individuals (parameter sets) by randomly modified parameters.
Network installation	installation process of a SIMPLORER network version.
NEWTON	simulation parameter; limits the number of iteration loops for the nonlinear iteration process.
Newton-Raphson-Algorithm	nonlinear iteration algorithm used in SIMPLORER.

Term	Description
Non-conservative node	connection of two or more non-circuit component terminals.
Normal mode	default display of the experiment tool without evaluation functions and storage options.
Object	element of the graphic model description linked to another WINDOWS application (OLE).
Object browser	special window for the display and browsing of the elements of a graphical model description.
Offline	when the simulation has finished in SIMPLORER.
Online	during the simulation in SIMPLORER.
Online graphic	online display of simulation results during the simulation run.
Online graphic output	display of a system quantity online during the simulation in an Display element or the ViewTool.
Option	optional SIMPLORER application; can be registered using the installation manager.
Output	definition of a specific quantity of the simulation model to be used as an output.
Password	letter and digit combination to access SIMPLORER.
Petri net	special form of a state machine.
Pipe	data channel for the data transmission between SIMPLORER modules.
Postprocessing	data evaluation and processing after a simulation run has finished.
Power	analysis method in the DAY Post Processor.
Pre process	specialized modules for information processing and parameter determination to define SIMPLORER model components.
Pre processor directive	special commands for the SML compiler to include files, extract macros from the model library, etc.
Presentation	special mode of the DAY Post Processor; allows the arrangement of results in an reusable form.
Print colors	colors used to print a system quantity from an active element or the ViewTool.
Project	organizational structure containing all files and information belonging to a simulation task.
Qualifier	references a system quantity or parameter of a SIMPLORER model component.
Quality criterion	characteristic value of a system quantity; used to determine the quality of an optimization run.
Queue	display the active and all waiting simulation runs.
Recombination	part of the genetic algorithm; generates new individuals (parameter sets) by crossing of two parent individuals.
Roll back	void a simulation step.
Sample time	step size for digital controller.
Schematic	SIMPLORER module for the graphical model definition.
Screen colors	colors used to display a system quantity online during the simulation in an active element or the ViewTool.
Section	part of the SML description containing model information for the SIMPLORER sub simulators.
Selection	part of the genetic algorithm; selects individuals of the active generation to be transferred to the next generation by specific selection criteria.
SIMPLORER program	SIMPLORER application launched from the SSC commander.
SIMPLORER settings	contains the settings for the program environment, paths, etc.
SIMPLORER tools	additional programs included in the software package but not necessarily integrated into the SSC commander.
Simulation backplane technology	SIMPLORER software architecture for data exchange and program control of several simulators in one environment.
Simulation model	graphical or textual description of a real system using modeling capabilities of a simulation system.

Term	Description
Simulation parameter	parameter used to control the simulation process.
Simulator	software for the analysis of the behavior of a system using a simulation model.
Simulator coupling	direct link of one or more simulators using the SIMPLORER simulation backplane technology.
Simulator interface	special software interface for the integration of external simulators into SIMPLORER.
.smd file	SIMPLROER Model Agent file; containing data of a model library.
SML	SIMPLORER Modeling Language, ASCII model description language of SIMPLORER.
.sml file	file containing the model description of a simulation run.
SML key word	special terms used to determine the sections of an SML description.
Smooth	analysis method in the DAY Post Processor.
Solver	algorithm for the computation of model components without the capability to roll back steps.
SSC commander	SIMPLORER application manager, project manager and communication interface.
.ssc file	SIMPLORER project file, containing all information about a project.
.ssh file	SIMPLORER schematic file, containing the graphical representation of a model and simulation data.
State	basic element of state graphs; defines properties and activities in a certain system state.
State graph	combination of states and transitions; modeling language for discontinuous systems.
State Graph Module (SGM)	SIMPLORER sub simulator to analyze simulation models described using state graphs basing on the PETRI net theory.
Status line	displays the present state of a SIMPLORER application.
Subsheet	a SIMPLORER sheet embedded into another SIMPLORER sheet, connected via pins, automatically creates a macro inside the .sml file.
Sub-simulator	SIMPLORER internally coupled simulator.
Sub-skeleton	file containing rules for the creation of a non-SIMPLORER description language.
Successive Approximation	optimization method of the Experiment Tool with automatic parameter variation and target function determination.
Symbol editor	SIMPLORER application for the creation or modification of a symbol.
Symbol level	group of drawing elements of an animated symbol displayed together depending on the input value or the user activity on a interaction pad.
Symbols	elements of the Schematic, placed from the model library; can be modified using the symbol editor.
Synchronization	update of a schematic from an older SIMPLORER version to the latest symbols.
Synchronization	detection of events in a state graph.
System quantity	any quantity computed by the simulator.
System simulation	simulation level, where models from different physical domains are simulated at the same time.
System variable	predefined variables inside SIMPLORER that cannot be used as a variable name or a specifier.
Task	analysis to be performed as part of an experiment; can contain several simulation runs.
Template	predefined structure for a SIMPLORER application.
TEND	simulation parameter that determines the simulation end time.
Text Editor	SIMPLORER application for the definition of models in SML language.
Time Function Module (TFM)	SIMPLORER sub simulator for the computation of time dependent functions.
Time limited	a license of the simulation software for a certain (limited) time period.

Term	Description
Time step	present time step size used to compute the next results vector.
Toolbar	bar in a SIMPLORER application to perform activities in the application.
Total fitness	the total fitness of an optimization run.
TR simulation	transient analysis of a model.
TR simulator	SIMPLORER simulator for TR analysis.
Transition	cross over condition between the input and output state(s) of a state graph, defined by a logical expression.
Transition component	basic structure of a state machine; comprises a transition and all of its input and output states.
Trapezoidal Formula	numerical algorithm used inside SIMPLORER.
Trend Analysis	analysis method of the Experiment Tool with automatic parameter variation and characteristic value determination.
UDMinit	section of the C/C++ interface for the model initialization.
UDMMain	section of the C/C++ interface containing the model computation algorithm.
User defined component (UDC)	model description for electrical components using a user defined C/C++ program; direct access to the solver matrix.
User defined model (UDM)	model description for nonlinear characteristics using a user defined C/C++ program.
User management	saves the workspace for individual users.
Version report	special mode inside the SIMPLORER help system to create a detailed information file about the present SIMPLORER installation.
VHDL	hardware description language for digital systems.
VHDL-AMS	extension of VHDL; hardware description language for digital and analog systems.
VHDL-AMS simulator	sub-simulator of the SIMPLORER system. It calculates simulation models described in VHDL-AMS.
ViewTool	program to display results online during the simulation outside the Schematic.
Window elements	windows in the workspace containing various information; can be turned on or off by the user.
Work flow	graphic representation of a sequence of activities that are performed on a certain data set.
Worst Case Analysis	analysis method of the Experiment Tool with parameter combination of all extreme values for the parameters and characteristic value determination.
YMAX	maximum value used for display in the online graphic using the View tool.
YMIN	minimum value used for display in the online graphic using the View tool.

A.2 Table of SIMPLORER Libraries

Tab	Description
Basics	The «Basics» tab provides electric circuit components, blocks, states, measuring devices, signal characteristics (functions to evaluate characteristics online during simulation), modeling tools (time functions, characteristics, equations), and components of physical domains.
Displays	The «Display» tab provides elements for visual online display of graphical and numerical simulation outputs during a simulation on the Schematic sheet.
AMS	The «AMS» tab provides electric circuit components, blocks, measuring devices, modeling tools (Time functions), and components of several physical domains modeled in VHDL-AMS.
Digital	The «Digital» tab provides components with common basic functionality used for simple digital circuits.
Add Ons	• Electric power components (power6.smd)* • Set of components suited for needs of the automotive industry (automotive6.smd)* • One-Dimensional Mechanical System Module (mechsim6.smd)* • Hydraulic components (hydraulic6.smd)* • Library with machine models (*_ist6.smd)* • Nonlinear transmission and linear transmission components (transfer6.smd) **Interfaces:** • ECE (equivalent ciruit extraction) Maxwell interface (interface.smd) • RMxprt interface (interface.smd) • Full-Wave SPICE interface (interface.smd) • Mathcad interface (interface.smd) • Matlab/Simulink interface (interface.smd)* **Compatibility:** • SIMPLORER 4: Semiconductors device level of former SIMPLORER versions • Digital: Digital Basic Gates in different modeling levels (NAND, NOR, ...). • Magnetic: Elements for nonlinear magnetic circuit modeling. • Function Blocks: Functional blocks libraries (Integrator, Frequency Doubler...). • Lines: Macros, based on the gamma equivalent circuit, to simulate line segments.
Tools	The «Tools» tab provides components to calculate coordinate transformations, components to connect conservative nodes from different natures, and components to connect different data types. You find the model descriptions in the SMPLORER online help.
Manufacturers	The «Manufactures» tab provides libraries with semiconductor device level models of different manufactures.
Users	The «User» tab provides a location to insert user-defined libraries.
Projcet	The «Project» tab provides libraries included in an opened project.

A.3 Common Conventions

Names of Components and Variables

User-defined names can be given to all components (internal components, C models, macros, nodes, ports and variables). Names may consist of letters, digits, and underscores and can have a maximum of 50 characters.

Vowel mutations (umlauts) and spaces are not allowed. All names are case sensitive. The first character must always be a letter.

The following are **not** allowed for names:

- SML notation keywords
- Simulation parameters
- System variables

Unit Suffixes of Numeric Data

Numeric data can be entered in Schematic component dialogs and in the Text Editor, using the following unit extensions:

Suffix	Value		SML		Examples
tera	10^{12}	E12	t	TER	*5e12, 5t, 5ter*
giga	10^{9}	E9	g	GIG	*1.4e9, 1.4g, 1.4gig*
mega	10^{6}	E6		MEG	*-1.4E6, -0.3meg, -0.3MEG*
kilo	10^{3}	E3	k	KIL	*1000, 1e3, 1k, 1kil*
milli	10^{-3}	E-3	m	MIL	*0.0105, 1.05E-2, 10.5M, 10.5MIL*
micro	10^{-6}	E-6	u	MIC	*0.000005, 5e-6, 5u, 5mic*
nano	10^{-9}	E-9	n	NAN	*40E-9, 40n, 40nan*
pico	10^{-12}	E-12	p	PIC	*100E-12, 100P, 100PIC*
femto	10^{-15}	E-15	f	FEM	*9E-15, 9F, 9FEM*

The comma is reserved for separating parameters in lists. The period (dot) is reserved as a decimal point. "M" is interpreted as 10^{-3}, not as 10^{6}.

A

SI Units

All units used from simulator immanent components are derived from the SI Units system.

Quantity	Unit Name	Symbol
Length	Meters	m
Mass	Kilograms	kg
Time	Seconds	s
Electrical current intensity	Amperes	A
Temperature	Kelvins	K
Voltage (derived SI unit)	Volts	V

Qualifier of Parameters

Components are characterized by different physical quantities. A resistor, for example, is represented by current and voltage in the simulation. System variables may be accessed by reading (to use the actual quantity in expressions or to create an output) or writing (influencing quantities). Use the following syntax to obtain access:

```
Name.Qualifier
```

Computations and outputs require access to system variables. The form and number of the qualifier depend on the corresponding component.

All qualifiers are case sensitive and must use capital letters (that is R.V not ~~R.v~~).

Predefined Variables

The simulator uses predefined variables for internal computation. If these variables are used in a model description, an error message or unexpected effects result. All predefined variables are case insensitive. See also “Using Simulation Parameters” on page 53.

Predefined variables cannot be used for names in a model description.

System constants (read)	`F`, `T`, `H`, `PI`, `TRUE`, `FALSE`, `SECM.ITERAT`, `FSTEP`
General simulation parameters (read/write)	`TEND`, `HMIN`, `HMAX`, `THETA`, `FSTART`, `FEND`
SML keywords	`MODELDEF`, `PORT`, `VAR`, `STORE`, `SIMCFG`, `OUTCFG`, `RESULT`, `SIMCTL`, `OUTCTL`, `RUN`, `INTERN`, `MODEL`, `UMODEL`, `COUPL`, `ELECTRICAL`, `MECHANICAL`, `REAL`, `INT`, `BIT`, `COMPLEX`

Predefined Constants

The simulator provides natural and mathematical constants that can be used in mathematical expressions within component dialogs or SML descriptions. The following table shows the available constants and their corresponding symbols:

Symbol	Value	Unit	Description	Variable
π	3.141592654	[/]	PI	MATH_PI
E	2.718281828	[/]	Euler number	MATH_E
ε_0	$8.85419\ 10^{-12}$	$C^2 \bullet Jm$	Permittivity of vacuum	PHYS_E0
μ_0	$1.25664\ 10^{-06}$	T^2m^3/J	Permeability of vacuum	PHYS_MU0
k_B	$1.38066\ 10^{-23}$	J/K	Boltzmann constant	PHYS_K
e	$1.60217733\ 10^{-19}$	C	Elementary charge	PHYS_Q
c	299 792 458	m/s	Speed of light	PHYS_C
g	9.80665	m/s^2	Acceleration due to gravity	PHYS_G
h	$6.6260755\ 10^{-34}$	Js	Planck constant	PHYS_H
ϑ	-273.15	°C	Absolute Zero	PHYS_T0

Equations, Expressions, and Variables

Equations consist of operands and operators. An operand can be any number or variable name. An operator compares or assigns a value. In expressions you can create and use variables as often as you want. A variable is defined when the variable name is used in an expression or for a parameter value within a component dialog. You do not need to define the variable in a specific assignment unless you want it to have a defined initial value.

Z:=Y+X; **X**,**Y**, and **Z** are the operands, and **:=** and **+** are the operators.

If the operants are complex numbers (for example in an AC simulation), the comparison operators (<, >, <=, >=) consider only the real part.

Assignment operators	:=	Assignment
	##	Delay operator combined with the action type DEL
Arithmetic operators	*	Multiplication
	/	Division
	+	Addition
	–	Subtraction
Comparison operators without synchronization	<	Less than
	>	Greater than
	<> or ><	Not equal to
Comparison operators with synchronization	This operator type forces the simulator to synchronize on the condition with the minimum step width.	
	<= or =<	Less than or equal to
	>= or =>	Greater than or equal to
	=	Equal to

Logic operators (must be surrounded by spaces)	`AND`	Logical AND (conjunction)
	`OR`	Logical OR (disjunction)
	`NOT`	Logical NOT (negation)

Standard Mathematical Functions

Mathematical functions consist of the function name and one or two arguments. An argument can be any number or variable name. An mathematical function applies the function, which it represents, to the argument(s).

`r:=FCT(x,y),r:=FCT(z)`

x, **y, and z** are arguments, z is a complex number, FCT is the function name, **r** is the result.

If the argument(s) are complex numbers (for example in an AC simulation), the functions RAD, DEG, DEGEL, MOD, INT, FRAC, LOOKUP consider only the real part.

Notation	**Description**	**Example**
`SIN(x)`	Sine, x[rad]	SIN(PI/6)=0.5
`COS(x)`	Cosine, x[rad]	COS(2•PI/3)=-0.5
`TAN(x)`	Tangent, x[rad]	TAN(PI/4)=1
`ARCSIN(x)`; `ASIN(x)`	Arc sine [rad]	ASIN(0.5)=0.524=PI/6
`ARCCOS(x)`; `ACOS(x)`	Arc cosine [rad]	ACOS(0.5)=1.0471=PI/3
`ARCTAN(x)`; `ATAN(x)`	Arc tangent [rad]	ATAN(1)=0.785=PI/4
`ARCTAN2(x,y)`; `ATAN2(x,y)` ATAN2=ATAN(y/x)	Arc tangent2 [rad] r=0 if x=0 and y=0; $-\pi \le r \le \pi$	ATAN2(.25,1)= ATAN(4)=1.325
`SINH(x)`	Sine hyperbola.	SINH(1)=1.175
`COSH(x)`	Cosine hyperbola.	COSH(1)=1.543
`TANH(x)`	Tangent hyperbola.	TANH(1)=0.762
`SQU(x)`	Square.	SQU(16)=16^2=256
`X^Y`	Power.	7^4=2401
`SQRT(x)`	Square root.	SQRT(9)=$\sqrt[2]{9}$=3
`ROOT(x,[y])`, y=2	n-th Root.	ROOT(27,3)=$\sqrt[3]{27}$=3
`EXP(x)`	Exponential function.	EXP(5)=e^5=148.41
`ABS(x)`	Absolute value.	ABS(-8.5)=|-8.5|=8.5
`LN(x)`	Natural logarithm.	LN(3)=$\log_e 3$=1.099
`LOG(x[,y])`; y=10	Common logarithm.	LOG(7,4)=$\log_4 7$=1.403
`INTEG(x)`	Integration of a variable from the function call until to the simulation end.	INTEG(var1)=$\int$var1 dx
`RE(z)`	Real part	RE(z)=5
`IM(z)`	Imaginary part	IM(z)=3
`ARG(z)`	Argument of a complex number in radians.	ARG(z)=0.53

Notation	Description	Example
	$z=a+bi=r(\cos\varphi+i\bullet\sin\varphi)=r\bullet e^{i\varphi}$ $z=5+3i=5.83(\cos 30.96^\circ+i\bullet\sin 30.96^\circ)$	
`SGN(x)`	Sign dependent value (-1, 0, 1). r=0 if z=0, 1 if Re(z)>0 or (Re(z)=0 and Im(z)>0), -1 otherwise.	SGN(3)=1; SIGN(0)=0; SIGN(-3)=-1
`RAD(x)`	Conversion from degrees to radians.	RAD(30)=PI/6=0.524
`DEG(x)`	Conversion from radians to degrees.	DEG(PI/2)=90°
`DEGEL(x[,y]); y=1`	Conversion from degrees electrical to seconds with respect to Hz.	DEGEL(180,50)=10ms
`MOD(x, [y]);` y=1	Modulus.	MOD(370,60)=10
`INT(x)`	Integer part of a value.	INT(2.5)=2
`FRAC(x)`	Fractional part of a value.	FRAC(2.5)=0.5
`LOOKUP(x,y)` x=Characteristic name y=X value	Access function to a characteristic.	LOOKUP(XY1.VAL,5)= Y value of the characteristic XY1 for the X value 5

`IF` (condition)
{ var:=1; }
[`ELSE IF` (condition)
{ var:=2; }
`ELSE`
{ var:=3; }]
	If-Else function to perform operations dependent on conditions.
	The ELSE IF and ELSE statement can be omitted.
		IF (t>=1)
		{ var:=1; }
		[ELSE IF (t>=2)
		{ var:=2; }
		ELSE
		{ var:=3; }

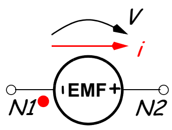
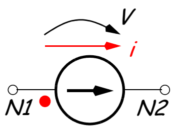
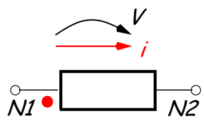
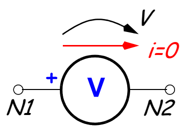
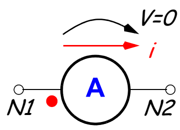
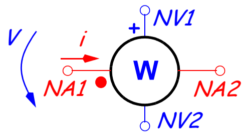
`DB(x)`	Conversion to Decibel (Available in the DAY Post Processor only.)	DB(8)=20•log(8)=18.062

When entering these functions, do not leave any space between the function argument, e.g., MOD and the open parenthesis mark.

If you define arguments for trigonometric functions, poles must be considered to avoid possible errors during the simulation.

Load Reference Arrow System of Circuit Components

The counting direction of current and voltage is marked by the red point or the plus sign at the symbol of electrical components.

<table>
<tr><th></th><th>Voltage Sources</th><th>Current Sources</th><th>Passive Components</th></tr>
<tr><td>Electrical</td><td></td><td></td><td></td></tr>
<tr><td></td><td>Voltmeter</td><td>Ammeter</td><td>Wattmeter</td></tr>
<tr><td>Meters</td><td></td><td></td><td></td></tr>
</table>

Network Configurations

In SIMPLORER only ammeters are allowed as controlling components for current controlled elements. These must be inserted properly in the controlling branch. If sources are part of mutual controlling sources in the circuit, stability problems may occur if the total gain of the loop is greater than or equal to one.

The following types of network configurations are invalid:

- Series connection of ideal current sources
- Series connection of inductors and ideal current sources
- Series connection of inductors with different initial values of current, $I0_1 \neq I0_2$
- Series connection of an inductor with an initial current value and an opened ideal switch or nonconducting system level semiconductor
- Parallel connection of ideal voltage sources
- Parallel connection of capacitors with ideal voltage sources
- Parallel connection of capacitors with different initial values of voltage, $V0_1 \neq V0_2$
- Meches which consist only of ideal sources (short-circuit)
- Open-ended branches

Actions in States

Action	Description	Syntax in Value
CALC	The variable is calculated at each simulation step and each transition from one state to another.	*var1:=100*t*
STEP	The variable is calculated at each **valid** simulation step.	*var2:=2*t*
CATI	The variable is calculated outside the state graph and before the calculation of the electric circuit.	*var3:=sqrt(t)*
SET	The variable is calculated only once at the moment of activation of the state.	*var4:=2.5*

Action	Description	Syntax in Value
DEL	Sets a delay. The variable is set to false at the moment of activation and set to true after the delay time.	var##time [s] *delation##10m*
DELRES	Deletes a defined delay variable.	del4
DIS	The variable value (and moment) is displayed in the simulator status window.	Name.Qualifier *dc.n*
TXT	The given text string is displayed in the simulator status window.	"Text String" "*State waiting*"
KEY	Sets a mark in the state graph by pressing a key.	*<A>*
STOP	Interrupts the simulation (can be continued).	No Parameters
BREAK	Finishes the simulation.	No Parameters
SAVE	Saves the active simulation status in a status file.	No Parameters

Basic Rules for the Proper Choice of Time Step

Correct simulation processing and results depend on the proper choice of minimum and maximum values for the integration step size. The smaller the maximum integration step size, the more correct the results, but the longer the processing time.

This means that, when specifying these minimum and maximum time step values, you need to compromise between accuracy and time. The basic rule of measurement "Not as precise as possible, but as precise as required" is also valid for a simulation. The following guidelines should help you prevent elementary mistakes in choosing the proper integration step width:

Model properties	Recommended
What is the smallest time constant (τ_{min}) of the electric circuit (R*C or L/R) or of the block diagram (PTn-elements)	$Hmin < \frac{\tau_{min}}{10}$
What is the largest time constant (τ_{max}) of the electric circuit (R*C or L/R) or of the block diagram (PTn-elements)	$Hmax < \frac{\tau_{max}}{10}$
Which is the smallest cycle (T_{min}) of oscillations that can be expected (natural frequencies of the system or oscillating time functions)	$Hmin < \frac{T_{min}}{20}$
Which is the largest cycle (T_{max}) of oscillations that can be expected (natural frequencies of the system or oscillating time functions)	$Hmax < \frac{T_{max}}{20}$
What is the smallest controller sampling (TS_{min})	$Hmin < \frac{TS_{min}}{5}$ $Hmax = TS_{min}$
What is the fastest transient occurrence (TU_{min}) (edge changes of time functions)	$Hmin < \frac{TU_{min}}{20}$
What is the time interval to be simulated (Tend)	$Hmax < \frac{TEND}{50}$

- Select the smallest of each estimated maximum and minimum time step for your simulation model.
- All the values recommended above are based on numeric requirements and experience and do not guarantee a successful simulation. Please consider the algorithm as a guideline.
- In case of doubt, decrease the maximum and minimum step size by dividing by 10, repeat the simulation and compare the results. If the second set of results (with the step size decreased) shows conformity with the first results, then the step sizes chosen for the first simulation were appropriate (remember that smaller values increase the simulation time).

If the number of iterations is identical with the defined maximal value (NEWTON) during the simulation, the model may be incorrect. The simulation monitor displays the active number of iterations. See also 13.4.1 “Displaying Monitors” on page 412.

A.4 Troubleshooting

Modeling

- Project must be copied before you can work with it
  Please remove the write protection from your files.
- Cannot connect elements on a sheet
  SIMPLORER schematic checks that the types of two pins to be connected are correct, e.g,. it is not possible to connect an electrical pin to a signal pin. So please check that you are connecting the proper pin types. If the connection is still impossible, please place the same element again.
- Connections are muddled
  Select the muddled wires, click the right mouse button and select «Disconnect» from the pop-up menu.
- Errors in the SML Syntax
  Using comma as decimal point is not allowed.

  Logic operators must surrounded by spaces.

  Predefined variables cannot be used for names in a model description!

  When using functions, there should be no space between the function argument and the open parenthesis mark.

  Names for variables and elements are case sensitive. Please check the style.
- Errors in the modeling
  Each independent circuit has to be connected to the ground node at least once. Only electrically reasonable circuits lead to correct simulation results. Voltage sources, current sources and switches are ideal components.

Display and Simulation

- No graphic is displayed
  It is possible that you did not define any outputs for a component (right mouse click and select Output) or you did not select the values to display in a Display Element.

  Please check the settings in EDIT>PROPERTIES. Check also, if the View Tool Window is reduced to a symbol in the Windows task bar. Restore it with the menu item «Restore».
- Simulation does not start
  Please check the simulation queue in the SSC Commander. Here you will see all active jobs. All simulations are placed in this queue and processed according the start time. Another possibility is that the simulator window is still open in the task bar. If so, please close it.
- Unexpected simulation results
  If the controlled sources are part of closed control loops in the circuit, stability problems can occur if the total gain of the loop is greater than or equal to one. In this case, (linearly or non linearly) controlled sources which directly access variables from the set or circuit equations should be used.
- The simulator computes always with the maximal step width (`HMAX`)
  If the simulation model only consist of blocks, the block diagram is computed (to each time step) with the maximal step width (`HMAX`).

A.5 Literature Reference

[1] Cooley J.M.,Tukey J.W.
An Algorithm for the Machine Calculation of Complex Fourier Series. Math. Comp. 19, 297-301 (1965) 443

[2] Massobrio, Giueseppe, and Paolo Antognetti
Semiconductor Device Modeling with SPICE. New York, McGraw Hill, 1993 155, 156, 157, 158, 159, 160, 163

[3] Roychowdhury, J. S., and D. O. Pederson
Efficient Transient Simulation of Lossy Interconnect. San Francisco, CA, ACM/IEEE Design Automation Conference, June 17-21, 1991 165

[4] Schrüfer E.
Signalverarbeitung, C. Hanser Verlag 1992 216

[5] Spiro, Hans
Simulation integrierter Schaltungen durch universelle Rechenprogramme,Verfahren und Praxis der rechnergestützten Simulation nichtlinearer Schaltungen, R. Oldenbourg Verlag München Wien, 1985 188

[6] Web site Electronics Laboratory of EPFL (Lausanne) at http
//legwww.epfl.ch/ekv/ 162

[7] Web site UC Berkeley Device Group at http
//www-device.eecs.berkeley.edu/~bsim3/ 160, 161

Index

Symbols

Numerics

A

B

C

Index

D

Index

E

H

I

Index

N

Index

Q

R

S

W

X

Z