IDEAS USER MANUAL

IDEAS 2024R1

ENGINEERED SUCCESS

Contact: ANDRITZ Inc.
Division: Automation & Digitalization
13560 Morris Road, Suite 1250
Alpharetta, GA 30004 / US
Phone: +1 770 640 2500
Fax: +1 770 640 2557
E-mail: simulation.support@andritz.com
andritz.com

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

Acronym	Description
ARMA	Autoregressive Moving Average
DCS	Distributed Control System
I/O	Input / Output
IPC	Inter-Process Communication
OLE	Object Linking & Embedding
OTS	Operator Training Simulator
P&ID	Piping and Instrumentation Diagram
PLC	Programmable Logic Controller

SECTION 1. IDEAS OVERVIEW

1.1 INTRODUCTION

IDEAS™* is a high-fidelity dynamic simulation tool for the engineer. It provides capabilities throughout the entire process plant life cycle.

Dynamic simulation is the simulation of real-world processes in which the results represent the material and energy flows of the process in a dynamic or time-varying fashion.

The fidelity of a simulation is a measure of its accuracy. Simulation models of process equipment strive to accurately represent actual equipment operation. The closer a model's results match the actual operation, the higher the fidelity of the model. IDEAS is a high-fidelity simulation software package encompassing the first principles of physics to describe physical, chemical, and electrical processes.

IDEAS is unique in that it provides a single tool for all the major phases of a process plant's life cycle:

- Conceptual Design
- Detailed Process Design
- Detailed Control Design and Analysis
- Control System Checkout
- Operator Training
- Operations Management

Traditionally, other dynamic simulation tools have been useful for one or two of the above-mentioned areas but not the others. IDEAS is the first high-fidelity simulation tool that can be used successfully in all these areas.

The graphical nature of IDEAS allows the user to structure a dynamic model by retrieving icon-based "objects" from various libraries and assembling them on a drawing-like worksheet. These objects generally have a one-to-one correspondence with actual process equipment, i.e., pumps, valves, tanks, controllers, etc. The user, therefore, can create a model by merely creating a P&ID-like picture of the process. Individual equipment characteristics such as pipe dimensions or pump curves can be specified by filling in Dialog Box information for each object. The Dialog Boxes can be accessed by double-clicking each object with the mouse.

The "objects" that are used to create a model reside in various libraries. These libraries are organized into logical groups such as Pipes & Valves, Pumps & Compressors, Analog Controls, etc. Every time an object is retrieved from a library and placed onto a worksheet; an "instance" of that object is created. Each instance of the same object is identical in terms of the code (programming) that is executed when the

* Trademark of the ANDRITZ GROUP.

model runs.

The objects are designed to "Plug and Play" with each other. They can be interconnected in a virtually infinite number of combinations to create complex models. The objects have color-coded connectors. The user simply has to connect the green connectors to the green ones and the red connectors to the red ones. These connections result in the connection protocol explained in Appendix A.

Once a model is created with objects retrieved from the libraries, and all the necessary information is entered into the Dialog Boxes, the simulation is ready to “run.” No other programming, "subroutine" connections, or other software manipulations are required. The flows in the pipes and valves as well as the associated pressures throughout the piping network are automatically solved, displayed in the dialog boxes, and are made available to transmitters, plotters, etc. This program structure thus evolves into a dynamic simulation tool that is high in fidelity and can be used by any process engineer.

1.2 TYPES OF MODELS

1.2.1 DYNAMIC MODELS: IDEAS GOLD

Dynamic models provide the information necessary for optimal process and control design. Process constraints and bottlenecks are often experienced during the transition between production rates or at start-up, shutdown, or upset conditions. In order to use a simulator for process control design and analysis, the simulator must be dynamic to model the time-varying parameters of interest. IDEAS provides all of these dynamic capabilities:

- First-principle dynamic simulation of mass, energy, and momentum in multi-component streams
- Highly accurate simulation of flow for both compressible and incompressible materials
- Object characteristics modeled to a high degree of detail
- Material Properties defined over a wide range of conditions
- True interactive operations allow model modifications and "what if?" analysis
- Rapid convergence for complex, recycled flow, real-world process models

1.2.2 MACRO MODELS: IDEAS BRONZE

Using IDEAS dynamic objects requires the user to specify details about the equipment that may not be known (or be relevant) during early stages of design. For example, in the conceptual design phases of a project, the user may need an overall plant energy and mass balance but may have no interest in pressure drops in pipes, specific pump curves, or valve sizes. IDEAS macro provides the tools to construct complex models of industrial processes using a number of generic objects such as mixers, splitters, separators, flow and pressure setters, supervisors, and a multi-equation object. The information about the process equipment required to make such models is much lesser than that required for the IDEAS dynamic objects.

The IDEAS macro objects are fully compatible with the IDEAS dynamic objects. Any model, which starts as a macro (steady state) model can evolve into a dynamic model as parts of the worksheet are replaced by dynamic objects, provided detailed information about the process and equipment becomes available.

As a result, a final process model may consist of some dynamic objects and some simpler macro objects.

1.3 DYNAMIC FINANCIAL MODELS (TIED TO PROCESS)

Since the ultimate aim for most industrial processes is profit, the cost of various portions of the process under varying production conditions is of great interest. IDEAS Financial provides the tools to track manpower, material, and energy costs dynamically, as the model runs, and to analyze the process to optimize profit. These costs can be accumulated, split, and reported.

With the cost per unit as the defined parameter and the number of units as the measured variable, the IDEAS Financial objects calculate the associated cost on a Per-Step and Total basis. The Operating Cost Delta object is designed to average and annualize profit and to determine the difference in cost between the current and previous simulation runs.

1.4 SCAN CLASS

A dynamic model is typically operated at a specified rate, or time step, called Delta Time. A scan through the model and solution of the piping network occurs every Delta Time. Often there is a special need for one or more objects to run slower or faster than the basic model's Delta Time rate. IDEAS allows the user to specify multiple scan rates, or scan classes, to accomplish this. The user, for example, may want to simulate a DCS analog controller that has a sampling rate of 0.2 seconds. The model can be set to execute every 0.1 seconds and the controller object in the model can execute every 0.2 seconds. This is accomplished using the Discrete/Continuous Executive object in the Executives library. For more information on this feature, see Appendix B.

1.5 SNAPSHOT

The Snapshot feature allows the user to save complete sets of model operating parameters. These operating states can then be loaded into the model, and the model can be started at this operating condition. This feature easily allows the user to return to operating states that may have taken many hours to attain the first time. More details on the Snapshot functionality can be found in the Executives library help text.

1.6 MATERIAL PROPERTY DATABASES

High-fidelity simulations require comprehensive material properties database capability. Each object must have access to enthalpies, densities, etc., for each component defined in the model. IDEAS provides several material properties databases for users in different industries.

- IDEAS MP database with steam tables
- Mining MP database with non-ideal mixtures (add-on)
- VMG MP database with VLE model (add-on)

More information on Material Properties objects (as well as how to create new component records) can be found in the Material Properties library help text.

1.7 PUMP AND VALVE SIZING CAPABILITY

IDEAS can be used for detailed process and control design. After the piping dimensions and elevations are entered into the model, the user can easily determine the size of pumps and valves in the model by using the automatic pump and valve-sizing feature. A pump and valve pair can be simultaneously sized to determine the head requirement for the pump and the Cv or Cg of the valve.

1.8 SOLVING PIPING NETWORKS

Very complex piping networks can be created with IDEAS. Networks can have many flow paths, recycled flows, intermittent reverse flows, and other characteristics that make an accurate flow and pressure solution difficult to achieve. Some simulators approach this problem by creating a very large matrix of equations or several artificially separated smaller matrices that must be solved simultaneously for every scan of the simulation. When things are working properly, this technique works well. However, by artificially separating the matrix into smaller matrices, a technique called "islanding", compromises are made that adversely affect model fidelity. If the model has areas in the piping system, which are extremely difficult to solve, as most complex models do, these solution techniques can diverge and produce invalid data. When these matrix techniques diverge, it becomes very difficult for the user to determine where the problem occurred since the entire solution is invalid. When dynamic models are solved, this approach often requires that steady-state models be constructed first to obtain proper initial conditions to prevent invalid solutions.

Other simulators rely exclusively on a "sequential modular approach" in which calculations are made step-by-step across the worksheet with equations solved by iterating through multiple passes through the worksheet. IDEAS has a pure steady-state mode available that uses this approach. For a fully dynamic simulation, IDEAS calculates solutions using a proprietary hybrid solver approach, combining the best of the simultaneous and sequential modular approaches. There is no need to select an overall numerical integration method, as IDEAS automatically adapts to the specific stiffness of the model without user intervention or knowledge of numerical techniques.

Each sequential modular pass through the worksheet is followed by a simultaneous series of iterations, which assures convergence of the worksheet at each time step (default = 1 second) to the accuracy specified by the user in the solver dialog. Every pass through the worksheet is followed by this global convergence calculation. In this manner, the pressure-flow network is solved sequentially through the model each scan, followed by the global simultaneous emulation convergence procedure. If any part of the model experiences a problem, it is isolated to that particular area of the model and it is easy to find via error messages and/or highlighting of the offending object.

The IDEAS Solver-Fluid Flow automatically solves very complex piping arrangements without resorting to "islanding" and can achieve the same convergence accuracy as the matrix methods (pure simultaneous solution methods)—yet is considerably easier to use. In addition to the global solver, complex individual objects solve their own internal equations using either sequential or purely simultaneous solution techniques as selected by the object developer. The total combination of approaches, coupled with automatic monitoring of global convergence without user intervention, makes the IDEAS proprietary hybrid solver very powerful (Appendix A provides more information about the IDEAS pressure-flow solution

technique).

1.9 CONTROL SYSTEMS

1.9.1 SYSTEMS EMULATION

Control function libraries allow the user to emulate vendor-specific control strategies within the simulator. This option allows the control design as well as the process design to be verified. Since the controls and the process model are part of the same simulation, a complete operating process can reside in one simulator PC on a design engineer's desk.

1.9.2 SYSTEMS DESIGN

IDEAS dynamic process simulation provides an excellent tool for control system design. Control system configurations and strategies can be designed and evaluated using a detailed model of the process connected to the emulated control strategy in the model itself. To optimize this process, IDEAS provides an optional library of specialized control design objects that perform data manipulation functions such as auto-correlation, plotting functions such as Bode Plots, estimation methods such as least squares fitting, various model structures such as ARMA, predictors such as k-step ahead prediction and control structures such as minimum variance control. For more information on these functions, refer to the detailed library documentation for the object.

1.9.3 SYSTEMS INTERFACE

Since IDEAS can be used in many different ways, it has the capability to connect to various other systems. For example, when used as an Operator Training Simulator (OTS), the simulator can communicate with a DCS console to provide the same operator interface that the operator would see in the real plant. This interface can be one of two different types.

If the controls for the process are emulated in the simulation, then the simulator can communicate with just a DCS console or PLC operator interface for operator training. If, on the other hand, the controls are to reside in the DCS or PLC hardware and the simulator can provide realistic process information and responses from the controls, the interface is between the simulator and the DCS controller hardware or the PLC hardware itself. This type of interface can be used for both operator training and control strategy checkout. A more thorough control system checkout can be accomplished by interfacing the simulator directly to the DCS/PLC input/output terminals using I/O Subsystem. This interface provides for the most thorough control system staging available anywhere and drastically reduces start-up problems and start-up time.

1.10 HELP TEXT

IDEAS help text is available from the Help button located in the lower left corner of each object dialog box. This help can be accessed by opening the object (double-clicking the icon) dialog box and then clicking on the Help button. The help provides the modeler with important information about the setup of the object and the necessary parameters required for the object.

SECTION 2. MENU COMMAND REFERENCE

This section describes the commands and menus available in IDEAS. It explains all the commands that appear in the menus and the circumstances in which the user might use them.

Some of the menus are hierarchical menus. A hierarchical menu has a submenu for one or more of its main menu items and is indicated by a triangle pointing to the right (▸). To use a hierarchical menu, drag the cursor down the menu to the desired command. As the command is selected, a second menu appears to the right of the first one. While still holding down the mouse button, slide the cursor to the right, then down the hierarchical menu to the desired choice, and let go of the mouse button.

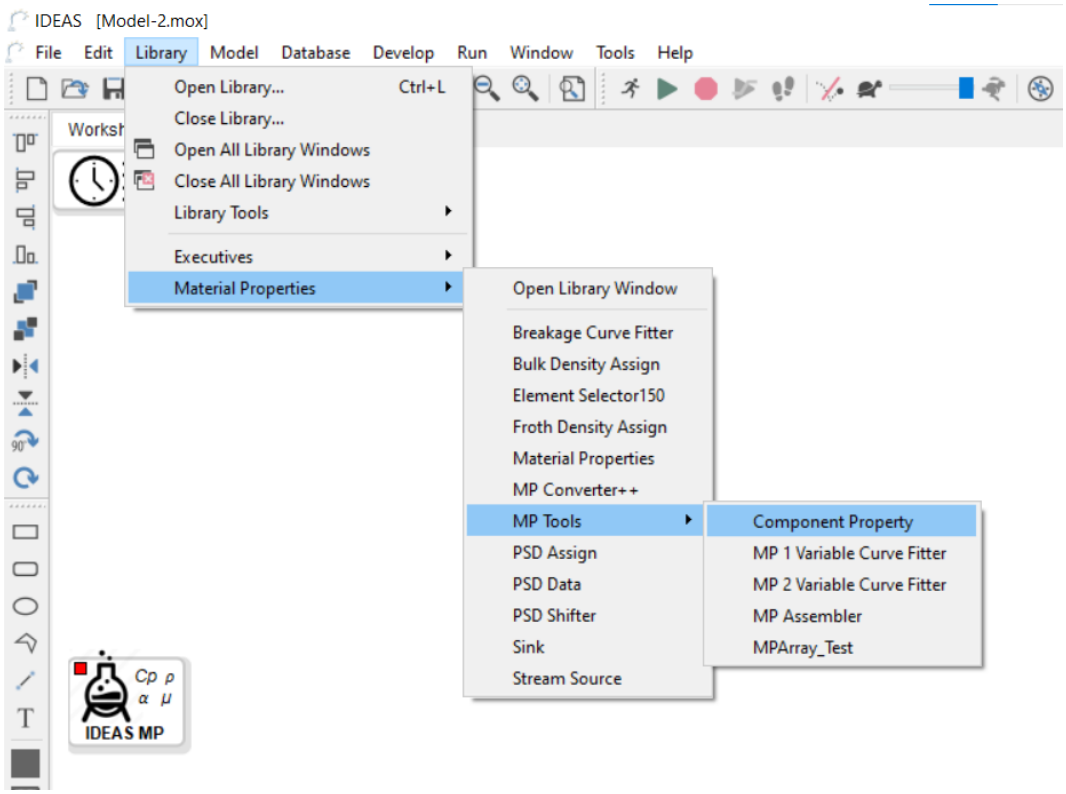


Figure 1: Hierarchical Menu

2.1 FILE MENU

The **File** menu opens, saves, and prints model files.

Note: The library files are opened from the Library menu described later. Most of the commands in this menu are the same as any other Windows program menu.

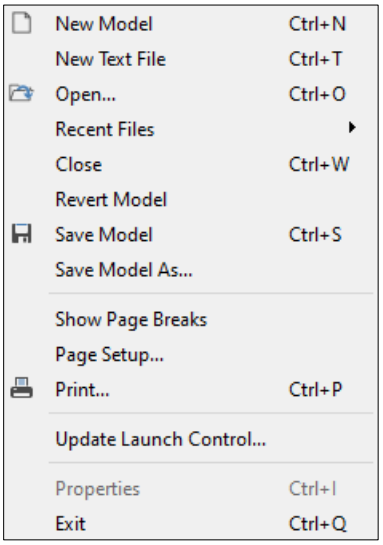


Figure 2: File Menu

2.1.1 NEW MODEL

Creates an untitled model worksheet.

2.1.2 NEW TEXT FILE

Creates an untitled text file. You can use this to create text files for any object to be able to read and write the data that uses a text file as input.

2.1.3 OPEN

Opens an existing model. As IDEAS opens the model, it also opens all libraries that are used by the model. Thus, there is no need to manually open libraries to run a model. Of course, if the library is already open, IDEAS does not reopen it.

If the model or the library has been moved since the model was closed, IDEAS may not be able to find the library on reopening the model. The message "Reading xxx..." will be displayed as IDEAS searches for the library. The user has to point to the location of the library in the standard file dialog box. Once the library file is found, highlight it, and select Open.

2.1.4 RECENT FILES

Lists the most recently opened files; new files will only be listed after they have been opened/saved.

2.1.5 CLOSE

Closes the active window—model worksheet, dialog, or text file. If there are any files with unsaved

changes, it first asks whether the user needs to save them.

2.1.6 REVERT MODEL

Becomes active after you make a change to a saved model. Reverts the model to the last version saved, discarding any changes made since then. IDEAS warns you before it completes this command.

2.1.7 SAVE MODEL AND SAVE MODEL AS….

Choose the Save option to save the file with the current name or Save As to save the file with a new name.

2.1.8 SHOW PAGE BREAKS

IDEAS draws a set of page boundaries on the model. These page boundaries show where page breaks occur if printed.

Note: The size of a page is dependent upon the settings in the Page Setup command under File menu. When Show Page Breaks is chosen, this menu item has a check mark next to it. To hide page breaks, select this command again.

2.1.9 PAGE SETUP

Sets the printing parameters for the printer being used. Choose this command after changing printers or whenever changing the printing parameters.

2.1.10 PRINT

Prints the model, its notebook, and/or the dialog boxes of the objects in the model.

2.1.11 UPDATE LAUNCH CONTROL

Selecting this command causes the currently active IDEAS application to be the application that opens when a model (.mox) or library (.lbr or .lbrpr) file is double-clicked.

2.1.12 PROPERTIES

Opens a window that gives information about the selected graphic object, block, etc.

2.1.13 EXIT

Terminates IDEAS application run. If there are any model files with unsaved changes, the user is prompted to save them.

2.2 EDIT MENU

The Edit menu contains the standard Cut, Copy, and Paste commands as well as other commands. Some

commands are described here. Others may be referred to in the *ExtendSim User Manual* in the Help folder within the IDEAS application folder.

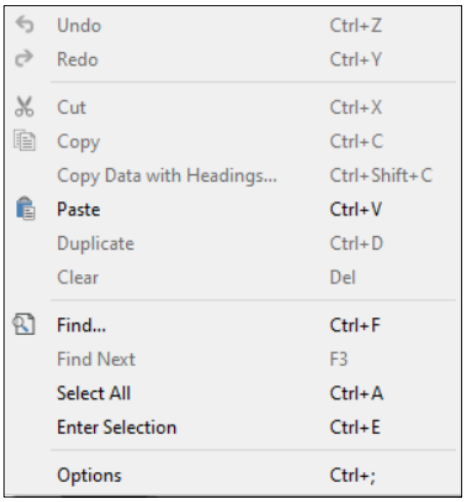


Figure 3: Edit Menu

2.2.1 UNDO

Reverses the most recent action. The user can undo commands, actions, and so on.

2.2.2 REDO

Reverses the Undo actions in a last-undone, first reverse order.

2.2.3 CUT

Removes the selected item (such as an object, text, or numeric data from a data table).

2.2.4 COPY

The user can copy a single object, a piece of text, a group of objects and text, graphical objects, or numeric values from a data table. This feature is useful for duplicating parts of a model as well as exporting to other applications.

2.2.5 COPY DATA WITH HEADINGS

Copies the table headings as well as the data from the selected data table.

2.2.6 PASTE

Pastes the contents copied earlier to the model. If the content contains text, an object, or a graphic item, the copied item is placed at the insertion point. If there is no insertion point, the item is placed in the upper

left-hand corner of the model.

2.2.7 DUPLICATE

Makes a copy of the selected item and puts it near the original item.

2.2.8 CLEAR

Removes the selected item.

2.2.9 FIND

Selects an object by its global Object ID (number). Object ID numbers are unique, permanent identifiers for objects. This command is useful on large models when looking for objects. The user can also extend the search by selecting Label or Name from the dropdown.

2.2.10 FIND NEXT

Finds the next instance of the object being searched.

2.2.11 SELECT ALL

Selects all the items such as all the objects in a model or all the text in a field. The items selected (objects and text, drawing items, and so on) depend on the selection tool chosen in the tool bar.

2.2.12 ENTER SELECTION

Causes the selected text to be the search string in the Find String dialog and the Find Next command. This is particularly useful for finding all instances of text that are already in the code of a block.

2.2.13 OPTIONS

User can customize the IDEAS features for the model.

Note: This command is different from the Simulation Setup command, which only affects a single model. The Options command controls actions for all models.

2.2.13.1 Options Menu - Model Tab

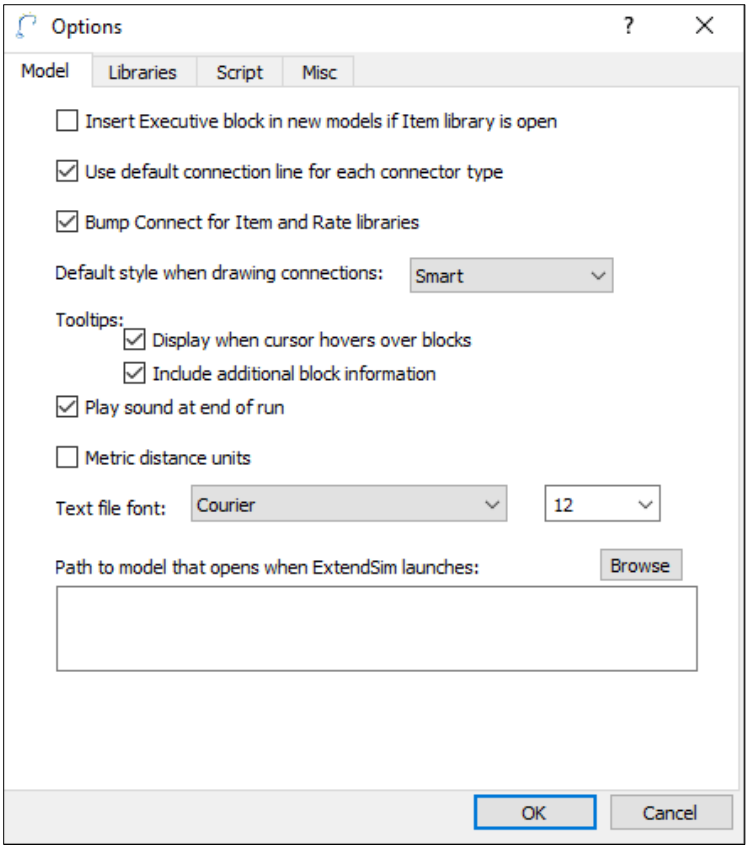


Figure 4: Model Tab

Use default connection line for each connector type: Always have the default line connection type selected.

Default style when drawing connections: Specifies the style for the connection lines.

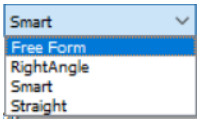


Figure 5: Default Styles Dropdown

Tooltips:

Display when cursor hovers over blocks:

a. When the cursor is hovered over a block, the block name, number, and the library name it belongs

to is displayed.

b. When hovered over a dialog box, the parameter name is displayed.

Play sound at end of run: The default sound is played at the end of every simulation.

Text file font: Sets the default font style and font size for text files.

Note: Some of the options are specific to the platform on which IDEAS is built; users need to leave these selections to the default.

2.2.13.2 Options Menu - Libraries Tab

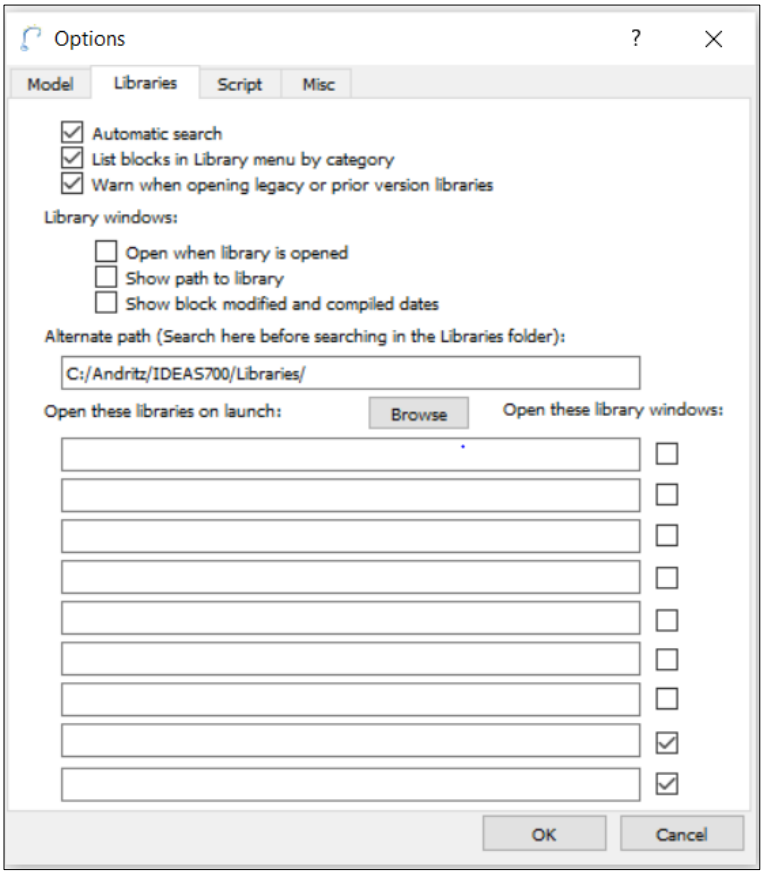


Figure 6: Options Menu - Libraries Tab

Automatic search: When a model is opened, IDEAS will automatically search and load the libraries that were saved with the model.

List blocks in Library menu by category: Allows the user to list the objects of a library category-wise.

Warn when opening legacy or prior version libraries: The warning appears if you use the Library > Open Library command to open a specific type of library: either a legacy (obsolete) library or an unconverted library developed in a prior IDEAS release.

Library windows:

Open when library is opened: Opens the corresponding library window whenever a library loads.

Show path to library: Causes the library window to display the path to the library below the library's name.

Show block modified and compiled dates: Displays the modified and compiled dates for each block in the library window.

Alternate path (Search here before searching in the libraries folder): The default path of libraries is displayed here.

Open these libraries on launch: Enter names of libraries to be automatically opened when IDEAS launches. Alternatively, user can also browse to the library location using browse button.

2.2.13.3 Options Menu - Script Tab

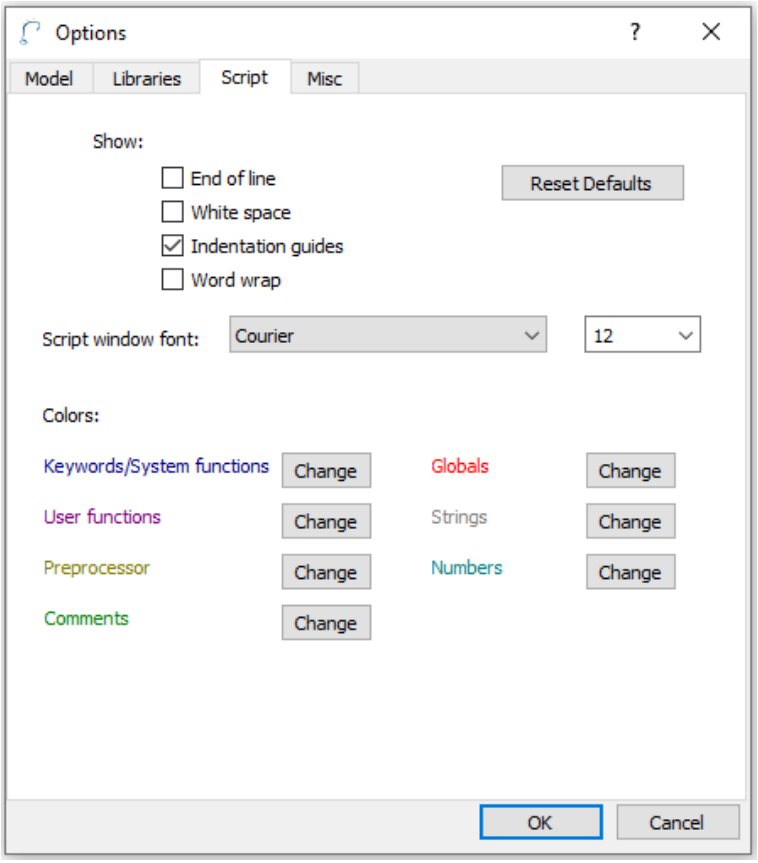


Figure 7: Options Menu – Script Tab

Show:

End of line: Shows the end of line characters as visible characters.

White space: Shows both tab and space characters as visible characters.

Indentation guides: Shows vertical lines to indicate the tab indentation of the code.

Word wrap: Also known as line wrap. When a line is full, this option automatically moves subsequent words to the beginning of the next line so that the text stays within the viewable window.

Script window font: Sets the default font style and font sizes

Colors: Shows the default colors for keywords, comments, and so forth, and provides Change buttons for selecting different colors.

2.2.13.4 Options Menu - Misc Tab

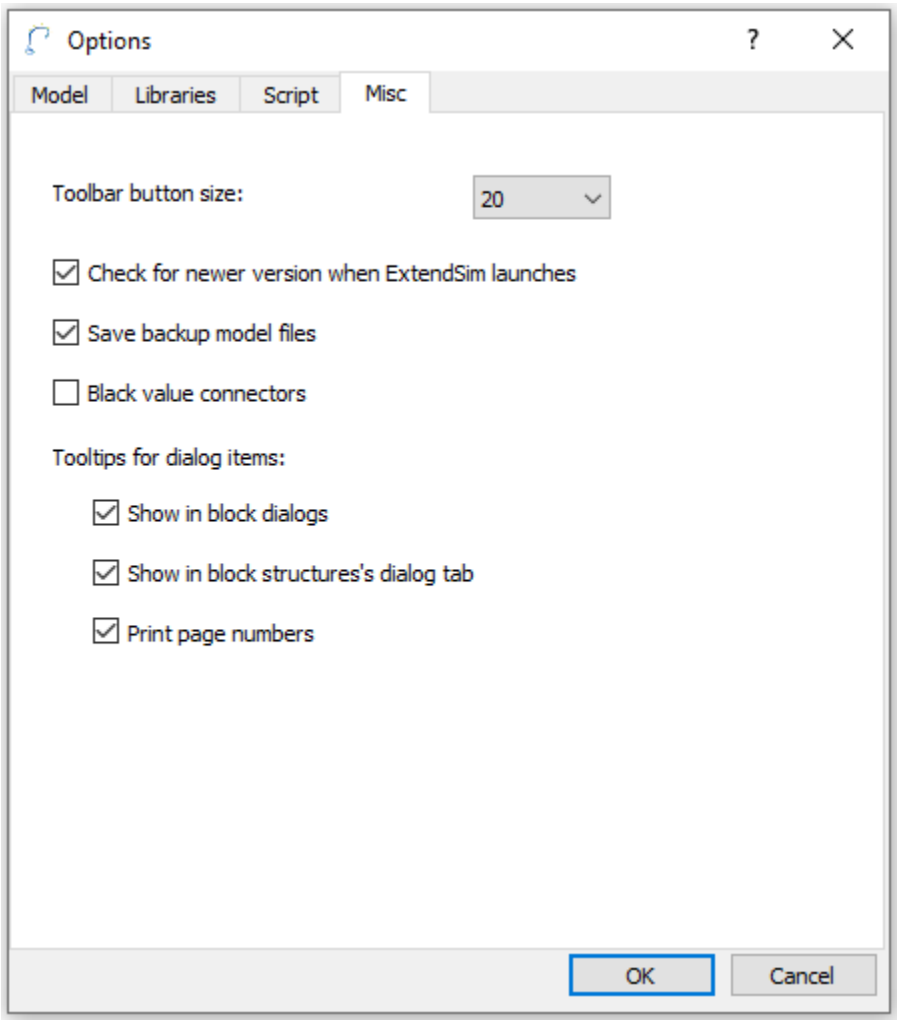


Figure 8: Options Menu – Misc Tab

Toolbar button size: Allows the user to change the size of the buttons in the toolbar.

Save backup model files: If a crash occurs during the save process, your original file could get corrupted. To protect against file corruption when saving, IDEAS first renames the previously saved version of the model as “ModelName.BAK”. To recover a model from a backup file, rename the backup file as ModelName.MOX.

Black value connectors: When checked, the value connectors become a distinct black for easy identification in the model.

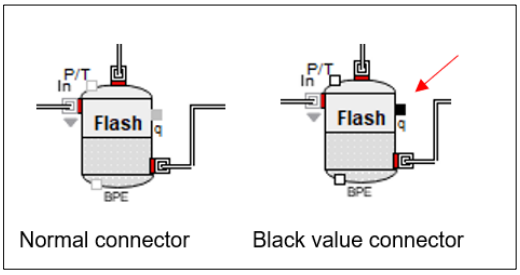


Figure 9: Black Value Connector

Tooltips for dialog items:

Show in block dialogs: Displays the variable/message dialog names when the cursor is hovered over fields in a block's dialog.

Show in block structure's dialog tab: Displays the variable/message dialog names when the cursor is hovered over fields in the Dialog tab of a block's structure.

2.3 LIBRARY MENU

IDEAS opens libraries automatically when models are opened. To open or close a library manually, use the Library menu. Add objects to the model by selecting them from the open libraries listed at the bottom of the Library menu. The first choice in each list is Open Library Window, which opens a window listing the objects in the library.

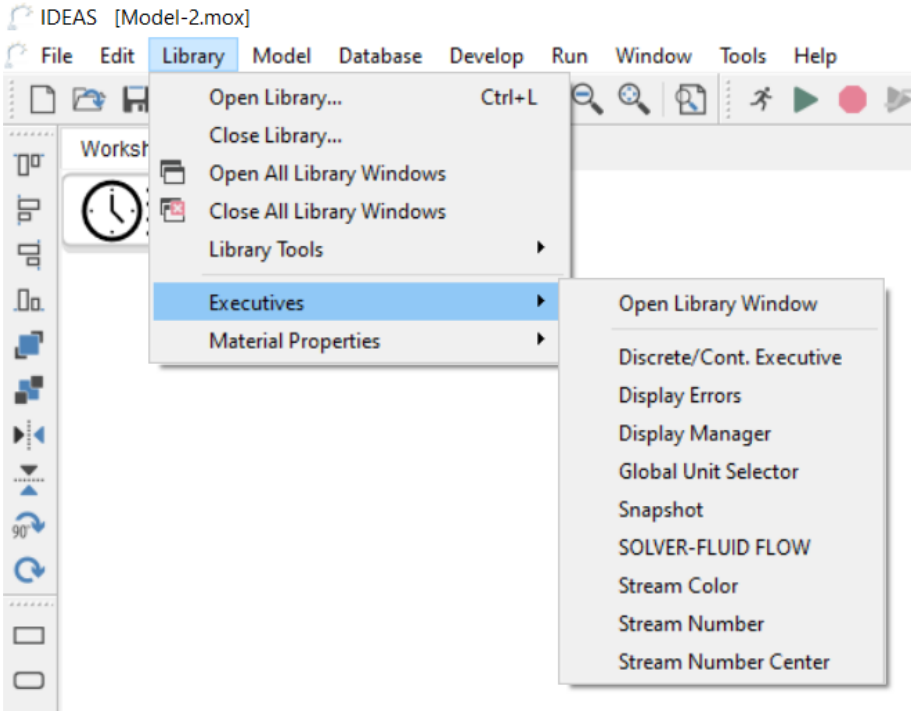


Figure 10: Library Menu

2.3.1 OPEN LIBRARY

Allows the user to browse, select, and open a library file. Upon opening a library, the library name is displayed in the Library menu.

2.3.2 CLOSE LIBRARY

Closes open libraries. This command displays a list of open libraries in a dialog box; user can select the libraries to be closed from the list and click Close Selected Libraries button. A SELECT ALL option is also available.

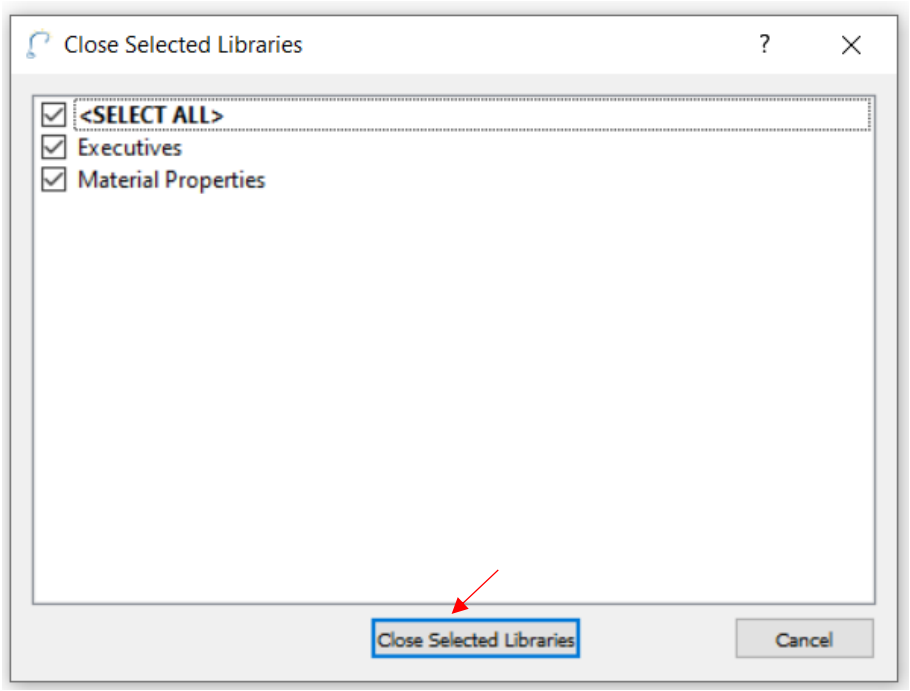


Figure 11: Close Libraries Window

Note: Libraries that are in use cannot be closed.

2.3.3 OPEN ALL LIBRARY WINDOWS

Opens all the libraries present in the library menu and docks the library windows to the right side of the IDEAS application window.

2.3.4 CLOSE ALL LIBRARY WINDOWS

Closes all the library windows that are currently open.

2.3.5 LIBRARY TOOLS

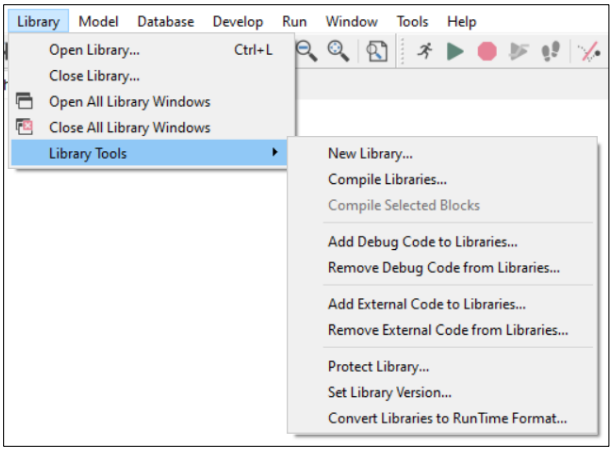


Figure 12: Library Tools

2.3.5.1 New Library

Creates a new library and opens a dialog for entering its name.

2.3.5.2 Compile Libraries

Opens a dialog for selecting the libraries to be compiled.

2.3.5.3 Compile Selected Blocks

Recompiles the selected blocks in a library. To activate this command, select blocks in the library window.

2.3.5.4 Add Debug Code to Libraries

Opens a dialog for selecting the libraries to be recompiled with the debugging code.

2.3.5.5 Remove Debug Code from Libraries

Opens a dialog for selecting the libraries to be recompiled without the debugging code.

2.3.5.6 Add External Code to Libraries

Opens a dialog for selecting libraries. This command moves the source code of each block in a selected library into a separate text file. This facilitates source code control.

2.3.5.7 Remove External Code from Libraries

Opens a dialog for selecting libraries. This command moves the external source code from separate text

files back to the respective blocks of a selected library.

2.3.5.8 Protect Library

After two (2) warning messages, opens a dialog for selecting the library to be protected. Protection removes the ModL source code from all the blocks in the selected library so that users cannot access the block code. This change is irreversible.

2.3.5.9 Set Library Version

Allows to set the long and short version strings for libraries developed by the user.

2.3.5.10 Convert Libraries to RunTime Format

Changes a copy of a selected library to Runtime format.

2.4 MODEL MENU

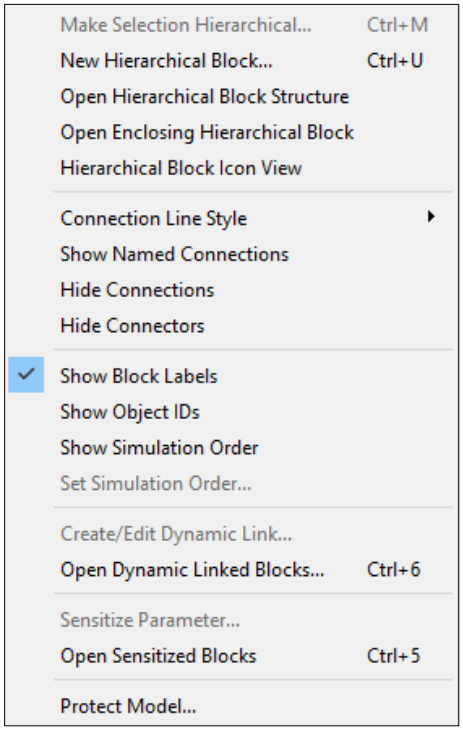


Figure 13: Model Menu

2.4.1 MAKE SELECTION HIERARCHICAL

Encapsulates the selected objects into a hierarchical block (H-block) and replaces them on the model with a single H-block.

2.4.2 NEW HIERARCHICAL BLOCK

This command allows the user to create a new H-block.

2.4.3 OPEN HIERARCHICAL BLOCK STRUCTURE

Allows the user to open the structure of an H-block and edit its Icon, Worksheet, or inbuilt Help.

2.4.4 OPEN ENCLOSING HIERARCHICAL BLOCK

Finds the H-block that contains the selected block. If the enclosing H-block is at the top level of the model worksheet, the H-block will be selected (shows as highlighted) on the worksheet. If the enclosing H-block is below the top level of the worksheet, the command opens the H-block's worksheet, showing the submodel that contains the selected block.

2.4.5 HIERARCHICAL BLOCK ICON VIEW

Places a miniature picture of its submodel on the selected H-block's icon.

2.4.6 CONNECTION LINE STYLE

Sets the format of the selected connections. These formats are described in detail in Section 3.21 (Connectors and Connections). The choices are:

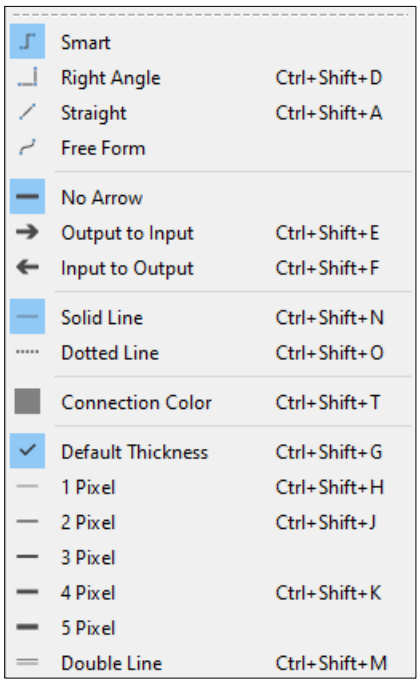


Figure 14: Connection Lines Menu

Note: If the user selects the dashed line at the top of the Connection Line Style dropdown, a moveable dialog window will appear containing the same list.

2.4.7 SHOW NAMED CONNECTIONS

Shows the connections between named connections. This feature is useful to show data flow in complex models with many named connections.

2.4.8 HIDE CONNECTIONS

Hides the connecting lines between objects. This is a cosmetic change that is mostly used to enhance presentations. Select the command again to show the connections.

2.4.9 HIDE CONNECTORS

Hides the connectors visible on blocks. This cosmetic change is mostly used to enhance presentations. Select the command again to show the connectors.

2.4.10 SHOW BLOCK LABELS

Shows the object labels below the objects in the model.

2.4.11 SHOW OBJECT IDS

Places the Object ID in square brackets (in blue) on each block in the model.

2.4.12 SHOW SIMULATION ORDER

Places the blocks' execution order number in square brackets (in red) on the blocks in the model.

2.4.13 SET SIMULATION ORDER

Opens a window for the user to set a selected block's execution order in the model.

2.4.14 CREATE/EDIT DYNAMIC LINK

Opens the Link dialog so a selected block's parameter field or data table can be linked to an internal data structure (IDEAS database or global array).

2.4.15 OPEN DYNAMIC LINKED BLOCKS

Opens the Find Links dialog that is useful for locating and examining linked dialog items or registered blocks.

2.4.16 SENSITIZE PARAMETER

If a dialog box parameter is selected, this command opens the Sensitivity Setup dialog box. The Sensitivity setup dialog box allows the user to set values for sensitivity analysis. Open Sensitized Blocks opens the dialogs of all blocks that have sensitized parameters.

2.4.17 PROTECT MODEL

Opens a window for protecting the model from any modifications other than changing dialog values. This can also be used to lock an H-block.

2.5 RUN MENU

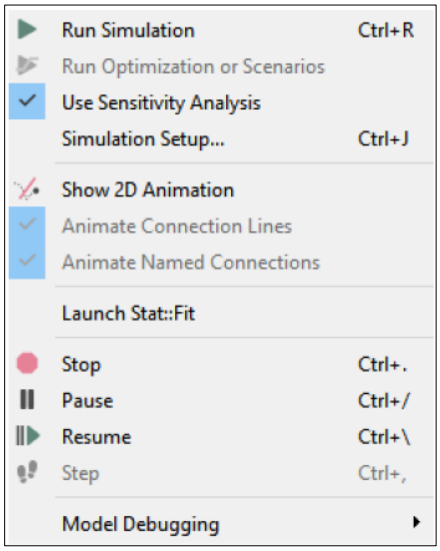


Figure 15: Run Menu

The Run menu lets you modify when and how your simulation runs, show animation, and generate model reports. A hierarchical menu at the bottom provides commands for debugging models.

2.5.1 RUN SIMULATION

Starts a simulation. This is the same as clicking the Run or button in the Model toolbar. The command first checks every block in the model to see that it has been compiled.

The icon of the Run Simulation button changes depending on which Run Mode has been selected in the Model toolbar. If the model has been set to run in the fastest (rather than multi-threaded) mode, the Run button will be .

Note that during the simulation run, the Run button in the toolbar changes to a Pause button while the

simulation is running and to a Resume button while the simulation is paused.

2.5.2 USE SENSITIVITY ANALYSIS

IDEAS uses sensitivity analysis settings when you run the simulation. Only enabled if a dialog parameter value has sensitivity settings.

2.5.3 SIMULATION SETUP

Opens a dialog for setting the start time, end time, and other settings for a simulation run as well as options for setting random number seeds.

2.5.4 SHOW 2D ANIMATION

Causes blocks in the model that have animation to animate when the simulation is run. Note that some blocks can show some animation, such as text on the icon reporting final values, even if Show Animation is not selected. You can also choose to animate along connection lines or between named connections, as discussed below.

2.5.5 ANIMATE CONNECTION LINES

This option controls whether or not 2D animation pictures will be displayed along connection lines in discrete event models. If on, blocks from discrete event libraries (such as the Item library), will display their item animations. If off, only animations on block icons will be displayed. This command requires that Show 2D animation be enabled and is only available for discrete event models.

2.5.6 ANIMATE NAMED CONNECTIONS

For named connections (where text labels, rather than connection lines, indicate the path of items), this option will cause the 2D animation picture to travel in a straight line between the two text labels. If this option is turned off, the animation picture will disappear when it has reached a text label and reappear at the matching text label. This command requires that Show 2D animation be enabled and is only available for discrete event models.

2.5.7 LAUNCH STATFIT (WINDOWS ONLY)

Opens the Stat::Fit application. This command is only enabled if Stat::Fit is installed. Stat::Fit is a Windows only product.

2.5.8 STOP

Stops the simulation. This is the same as the Stop button  in the toolbar. As an alternative, hold down the Ctrl (Windows) or Command (Mac OS) key while pressing the period (.) key. In any case this will cause a window to appear where you can choose to continue the run or end it.

2.5.9 PAUSE

Halts the simulation temporarily. This is the same as the Pause button in the toolbar. Note that the Pause button replaces the Run button in the toolbar once the simulation starts running. Once the simulation is paused, the Resume button appears until the simulation is resumed. To restart the simulation, give the Resume command, below, or click the Resume button.

2.5.10 RESUME

Restarts a paused simulation. This is the same as the Resume button in the toolbar. Once the simulation has resumed, the Pause button appears in the toolbar until either the simulation gets paused or the run finishes.

2.5.11 STEP

Steps the simulation run depending on which option (Step Each Block or Step Entire Model) is selected in the Run > Model Debugging menu. This is the same as the Step button in the toolbar.

2.5.12 MODEL DEBUGGING

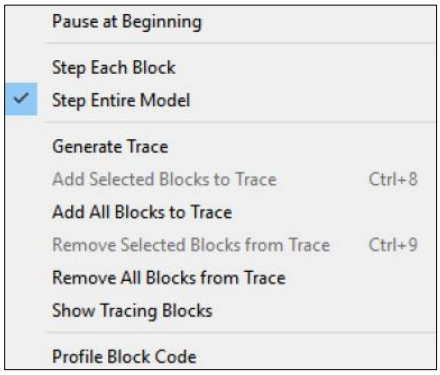


Figure 16: Model Debugging Options

This hierarchical menu, shown above, facilitates finding a modeling problem.

2.5.12.1 Pause at Beginning

If enabled, pauses the simulation after it starts so that you can step through the run from step zero. The first pause occurs before the first step but after the initial model processing (initialization, error checking, etc.) While the simulation is paused, the word ***Paused*** appears in the model's status bar.

2.5.12.2 Step Each Block

Controls the behavior of the Step command or button so that you can step through a simulation block by block. This is only active when the Pause button or command has been activated.

2.5.12.3 Step Entire Model

Controls the behavior of the Step command or button so that you can step through an entire cycle of all the blocks in the model. Each Step command starts at the selected block and continues the simulation run until the execution order returns to the original block. Only active when the Pause button or command has been activated. This is a good way to examine what happens in the intervals between when a block is called.

2.5.12.4 Generate Trace

The Trace commands are used to generate a Trace file of the values for each selected block in the active model worksheet at each step of the simulation. IDEAS will prompt for a name for the new trace file when the model starts running. The content of the trace file depends on which blocks have been selected to be included.

2.5.12.5 Add Selected Blocks to Trace

Causes blocks that have been selected in the active model worksheet to be included in the trace file. Trace files are generated if Generate Trace is checked when the model is run.

2.5.12.6 Add All Blocks to Trace

Causes all blocks in the active model worksheet to be included in the trace file. Trace files are generated if Generate Trace is checked when the model is run.

2.5.12.7 Remove Selected Blocks from Trace

Causes blocks that have been selected in the active model worksheet to be removed from the trace file.

2.5.12.8 Remove All Blocks from Trace

Causes all blocks in the active model worksheet to be removed from the trace file. Use this before starting a new type of trace.

2.5.12.9 Show Tracing Blocks

Causes blocks that have been included in a trace to show the word *Trace* on them in the model worksheet.

2.5.12.10 Profile Block Code

Generates a text file showing the percentage of time that each block spent executing during the simulation run.

2.6 WINDOW MENU

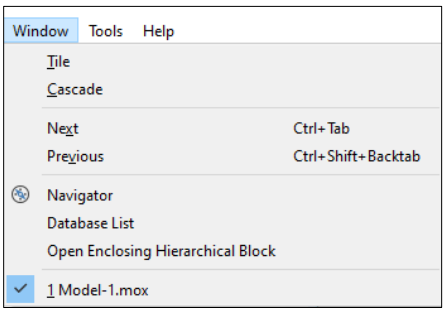


Figure 17: Window Menu

For the Windows operating system, the Window menu lists the standard Tile, Cascade, Next, and Previous commands. The menu also lists open IDEAS windows if opened, i.e., models, dialogs, and library windows, at the bottom. To bring a window to the top of your workspace, select it from this menu.

2.6.1 NAVIGATOR

Opens or brings forward the Navigator for the active model. This is the same as choosing the Navigator button in the toolbar. The Navigator is an explorer-type interface for a model worksheet, showing all blocks and hierarchical layers.

2.6.2 DATABASE LIST

Opens or brings forward a window showing the databases (if any) used in the model. This is the same list of databases that is shown at the bottom of the Database menu.

2.6.3 OPEN ENCLOSING HIERARCHICAL BLOCK

Finds the hierarchical block that contains the selected block. If the enclosing hierarchical block is at the top level of the model worksheet, the hierarchical block will be selected (shows as highlighted) on the worksheet. If the enclosing hierarchical block is below the top level of the worksheet, the command opens the hierarchical block's worksheet, showing the submodel that contains the selected block.

2.7 TOOLS MENU

Also refer to Section 3.12 Toolbar for more detailed explanation about the Tools options.

2.8 FILE TOOLBAR

Figure 18: File Tools

The standard buttons for opening a new model, opening an existing model, saving, and printing.

2.8.1 EDIT TOOLBAR

Figure 19: Edit Tools

The standard Cut, Copy, Paste, Undo, Redo, and Find buttons. Between the Undo and Find buttons are buttons for zooming in and out, returning to normal size, and zoom to fit.

Hint: On the model worksheet, hold down Ctrl while moving the mouse's scroll wheel to zoom in and out.

What happens when you use Cut, Copy, and Paste depends on the cursor that is used. For example, using the Block/Text cursor to frame-select and copy disparate items (blocks, text, graphic objects, pictures, etc.) will only copy the blocks and the text. Using the All Objects cursor would copy all the items.

2.8.2 MODEL TOOLBAR

Figure 20: Model Tools

Controls simulation and animation behavior and opens the Navigator. The first button is the Run Mode button. This option toggles to fastest run mode (shown above) when running a single model one or more times and uses the multi-threaded run mode when running multiple models at the same time.

The Run Simulation button is either a single green arrow if the Run Mode (discussed above) is set to fastest run or multiple arrows if the Run Mode is set to multi-threaded. Use the fastest run for single models, even if running them multiple times. Use multi-threaded if you are running multiple models at the same time.

The Run Simulation button changes to a Pause button while the simulation is running and to a Resume button if the simulation is paused.

The Animation Slider and the Faster and Slower Animation icons apply only to 2D animation.

The last button on the right, the Navigator, presents an explorer-type representation of the model.

2.8.3 CURSORS TOOLBAR

Figure 21: Cursor Tools

Displays the Block/Text, Graphics, Clone, and All Objects cursors. Also shows the position of the cursor on the model worksheet and on the Icon and Dialog tabs of a block's structure.

Depending on which cursor is used, different behaviors will occur. For example, using the Block/Text cursor to frame-select and copy disparate items (blocks, text, graphic objects, pictures, etc.) will only copy the blocks and the text. Using the All Objects cursor would copy all the items.

2.8.4 TEXT TOOLBAR

Figure 22: Text Tools

For setting the font and its style (such as Bold or Underline) and size.

- For model-wide use, set the font and format before creating any text boxes
- Create a text box by double-clicking on the model worksheet or by selecting the Text Box button in the Shapes tool.
- The Align Text button, to the right of the font styles, cycles through options for the selected text: left adjusted, centered, or right adjusted.
- By default text is drawn with a transparent background. Toggle the Transparent Background button, shown to the right of the Align Text button above, to cause the background of the selected text to be an opaque white.

For text files, which by nature don't retain font information, set the default font and size in the Edit > Options > Model tab. That option does not apply to the text boxes discussed above.

2.8.5 LIBRARY WINDOWS TOOLBAR

Figure 23: Library Window Tools

These two tools are located by default at the far right of the toolbar, over the library window docking area. Use them to open or close the library windows for all open libraries.

Opening or closing a library window does not open or close the associated library. See the Library menu for opening and closing a library.

2.8.6 ALIGNMENT TOOLBAR

Figure 24: Alignment Tools

Use these tools to manipulate graphic objects such as shapes, text, and lines. The buttons allow you to align selected text and graphics, cause an overlapping object to be sent to the back or brought to the front, flip a selection of multiple objects, and rotate a selected object 90 degrees from where it is or (Free Rotate) a specified number of degrees from zero.

Depending on which cursor is used, different behaviors will occur. For example, using the Block/Text cursor to frame-select and align disparate items (blocks, text, graphic objects, pictures...) will only align the blocks and the text. Using the All Objects cursor would align all the items.

The Flip buttons only flip the relative positions of text blocks that have been selected together. They do not result in the displaying of the reverse image of a graphic. Use an application such as Photoshop to create reverse images of graphics.

2.8.7 SHAPES TOOLBAR

Figure 25: Shape Tools

The shape toolbar is used for drawing and coloring shapes, lines, text, and for adding borders to shapes and text.

To get a square or circle, right-click a Rectangle or Oval shape to access its Properties. Then set the object's width and height to be the same value.

By default, text and shapes, other than lines, do not have a border. Before selecting a border color, choose a border width using the Line/Border Thickness button.

2.8.8 DIALOG ITEMS TOOLBAR

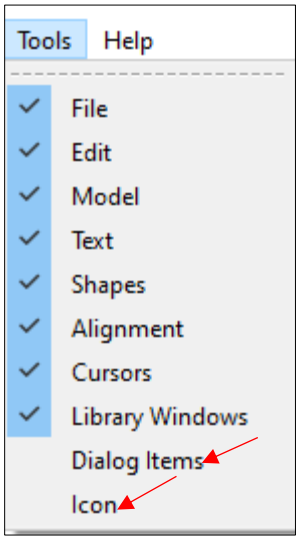


Figure 26: Enabling Dialog Items and Icons Toolbars

The Dialog Items and Icon toolbars are displayed only if the user checks them in the Tools menu. Once

enabled they appear to the right side of the IDEAS worksheet.

Figure 27: Dialog Items Tools

Inserts dialog items (buttons, checkboxes, popup menus, controls, etc.) used when creating a block's dialog. Since this set of buttons is only used when building or modifying blocks, the tool is unchecked by default.

2.8.9 ICON TOOLBAR

Figure 28: Icon Tools

Inserts connectors and animation objects used when creating a block's icon. Since this set of buttons is only used when building standard or hierarchical blocks, the tool is unchecked by default. The connectors are described in Section 3.21 (Connectors and Connections).

2.9 HELP MENU

The Help menu provides information about the application on the Windows platform.

2.9.1 ABOUT IDEAS

Displays the version of IDEAS as well as the user information in a splash screen.

2.9.2 IDEAS HELP

Shows a list of library-wise help topics. Select a topic from the list and double click for more information.

All the other options under Help menu when clicked takes the user to the corresponding ANDRITZ website for downloads and training information.

SECTION 3. MECHANICS

3.1 IDEAS FILES

There are two types of IDEAS files: models and libraries. A model shows the representation of objects and connections. A library is a repository of objects. When adding an object to a model, select it from a library.

Models are opened and closed from the File menu. When the user opens a model or creates a new model, the model window appears on their screen as a normal file window.

Libraries, on the other hand, are opened and closed from the Library menu. Libraries have their own library windows. They are also displayed at the bottom of the Library menu.

IDEAS can also use text files. These files can be read by any word processor and most other text editors.

3.2 IDENTIFYING THE MODEL COMPONENTS

The IDEAS dynamic simulation model contains many parts. The model consists of objects (seen as icons on the worksheet) that are graphically connected to resemble a process. Objects are sometimes referred to as blocks. Embedded within each object may be a number of differential and/or algebraic equations and data that describe the behavior of the object. The objects are stored based on their functions in appropriate libraries. A collection of objects for a model that resembles a P&ID is called a model worksheet. The most important parts of the model are the objects, the libraries from which the objects are loaded, the dialog boxes associated with each object, the connectors on each object, and the connections between the objects. The simulation itself is a series of calculations and actions that proceed along the path of the connections repeatedly. Each repetition is called a step or an event; the model can run up to $2*(10^9)$ steps.

The objects are processed according to the simulation order which can be selected in the *Simulation Setup* option (**Continuous tab**) in the **Simulation Setup** window of the **Run** menu.

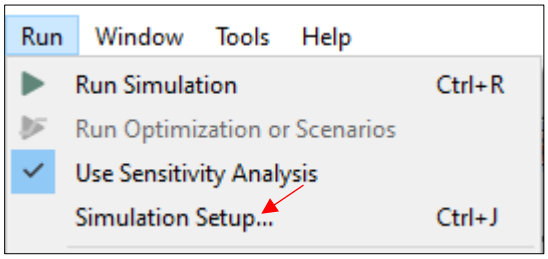


Figure 29: Simulation Setup Option

The options for setting up the Simulation order are shown in Figure 30 below.

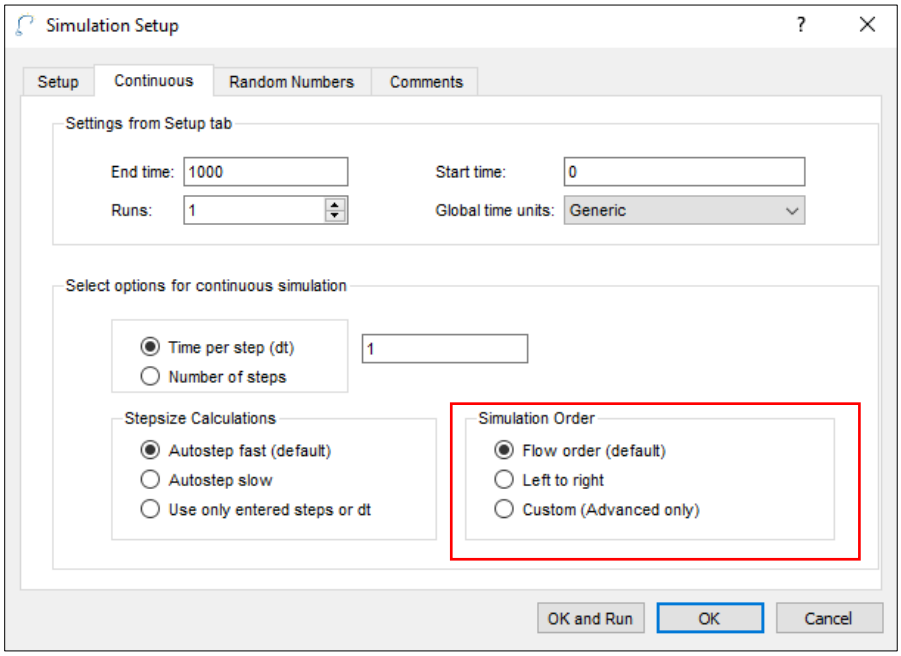


Figure 30: Setting Up the Simulation Order

After the first step, the simulation repeats itself. The user instructs IDEAS on the time per step and the number of steps in the **Simulation Setup** window under the **Run** Menu. This is explained in Section 3.23.1 in more detail.

When saving a model on the computer, the names of the objects as well as the locations of the libraries that the objects come from are saved. When opening a model, the required libraries are automatically loaded.

3.3 IMPORTING AND EXPORTING WITH TEXT FILES

A text file is an unformatted file created by a spreadsheet, database, or word processing program. The general method for creating a text file in those programs is to choose Save Text Document As from the File menu and specify the file format to be Text (.txt). Text files contain written text and/or formatted numerical data. There are many situations where the user wants to read and write text files (also known as ASCII files). These kinds of files can also be imported or exported in IDEAS.

For example, the user may want to share data with a database or spreadsheet program such as MS Excel. IDEAS can save files as text documents or perform an Inter-Process Communication (IPC) to a database or spreadsheet. The Import Database and Export Database commands in the Database menu can be used to read and write text files in dialog boxes. The Import Tables to Database and Export Selected Tables options can be used to read and write tables.

3.4 CUT, COPY, AND PASTE

The user can copy and paste portions of the model and data from IDEAS text entry fields and data tables. Objects and model sections can be pasted into graphics applications, word processors, or other IDEAS models. Data from text entry fields and data tables can be pasted into any other application such as word processors and spreadsheets. IDEAS uses the Cut, Copy, and Paste commands just like other programs.

When copying and pasting objects within IDEAS, the selected portion also holds object parameters and connections. This procedure allows the user to paste portions of the models, including the variables in the dialog box, to another section of the model or to other models. If the selected portion has objects in it when moving to another application, the contents are made available as pictures (stored in a graphics format). This feature is useful if the user wants to show parts of the model in reports or presentations. When pasting a picture (PICT objects or bitmapped) from other programs into IDEAS, it becomes a drawing object, which can be resized and repositioned.

To copy a graph/plotter image, right click on the plot and select Copy Graph Image. The user can copy data from a plotter's table by selecting the data and choosing the Copy command. In this case, IDEAS puts the data in text format. Similarly, the user can copy text from dialog boxes by selecting it and using the Copy command. Note that the user can select data from tables and copy it to other applications such as spreadsheets–this option makes it very easy to export data to other applications.

The user can copy pictures into IDEAS in color or black-and-white to use in the model windows, hierarchical windows, or in the icon in an object's structure window. The user can create the picture in a painting or drawing program such as CorelDraw or Paintbrush and copy it. Then it can be pasted into an IDEAS window with the Paste command from the Edit menu. Pictures copied into IDEAS - as objects created with IDEAS drawing tools – by default go behind IDEAS objects and text.

3.5 SCREEN CAPTURES

To copy parts of a model such as the entire plotter window or a dialog box, use a screen capture program to save the selection in a graphics file. Some screen capture programs will save just a portion of the screen.

3.6 INTER-PROCESS COMMUNICATION

The means by which applications communicate with each other and share data are collectively known as inter-process communications. Some communication methods have previously been discussed in this manual. The following section focuses on more extensive communication methods that allow IDEAS to directly communicate with other applications while the simulation is running. This allows IDEAS to work in conjunction with other applications on a wide variety of tasks.

Inter-process communication provides a standard way in which one application can directly communicate with another. In IDEAS, the user can incorporate IPC into the models using objects from the DDE library, or the user can program using the OLE/COM functions.

Communicating applications are typically categorized as clients or servers. A client application requests a service from some other application and a server application responds to the client's request. Many applications, such as IDEAS, act as either a client or a server depending upon the circumstances.

As a client application, IDEAS can request data and services from a server application.

As a server application, IDEAS can receive and execute ModL code that is sent from a client application. This means that other applications can instruct IDEAS to perform any task supported by the ModL language.

IDEAS supports OLE/COM functions to facilitate ActiveX controls and COM.

IDEAS is cross-platform compatible - the only difference between platforms is how inter-process communication is implemented.

3.7 LINKING DATA FROM ANOTHER APPLICATION INTO IDEAS

The user may wish to use data from another application as input parameters for objects in the model. To do this, create a dynamic link between the data and any parameter field or data table in any of IDEAS' object dialogs as follows:

In the server application, copy the data to the clipboard using the other application's Copy command. In IDEAS, select the parameter field to which the user wishes to copy the data.

Note: If the user is copying to a data table, the entire range of cells that need to be linked must be selected.

3.8 LINKING DATA FROM IDEAS INTO ANOTHER APPLICATION

The user may decide to export data from the model to another application for further analysis or presentation.

3.9 SERIAL PORTS

Please contact IDEAS Customer Support (see Section 5) if the user needs to interface the model through the serial port.

3.10 OTHER DEVICES, EXTERNAL CODE SEGMENTS, AND DLLs

Please contact IDEAS Customer Support (see Section 5) if the user needs to interface the model with outside hardware devices or to run external code resources.

3.11 COPYING PLOTTED INFORMATION

To copy a graph/plotter image, right click on the plot and select Copy Graph Image. To copy the data, the user must select it, and then choose the Copy command.

3.12 TOOLBAR

The tools in the toolbars at the top and both sides of the screen are used when interacting with the model window. The tools available when a worksheet is active are as shown in Figure 31.

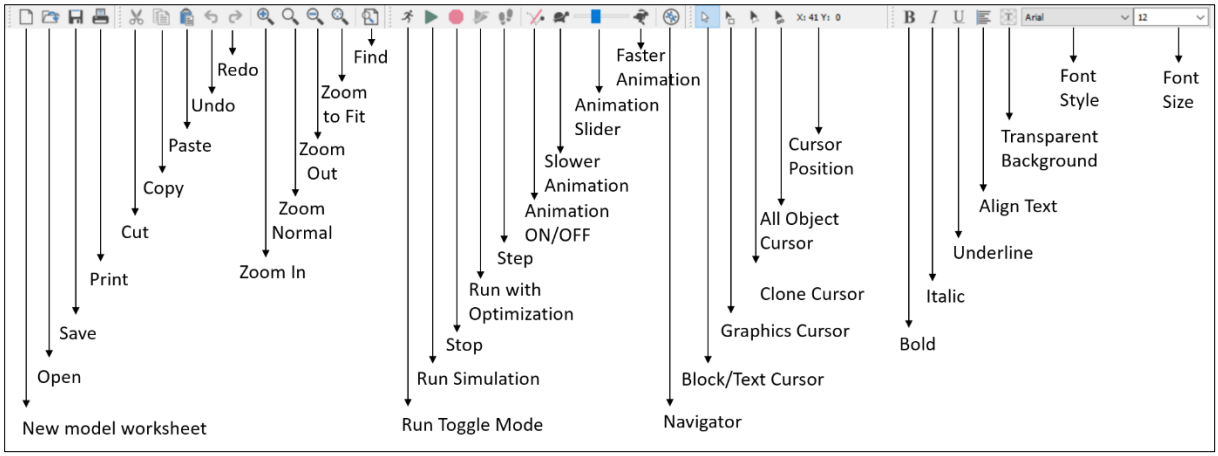


Figure 31: IDEAS Toolbar

There are different types of toolbars that serve various functions such as the file toolbar, edit toolbar, model toolbar, text toolbar, shapes toolbar, and alignment toolbar. These toolbars can be torn away and docked at locations convenient to the user such as the sides and the top of the window.

3.12.1 FILES

The first group is a standard set of buttons that deal with files. The user opens a new model worksheet by clicking on the New Model worksheet button. The Open button will open existing worksheets or text files. The Save and Print buttons save and print the active worksheet or text file, respectively.

3.12.2 EDITING

The second group is a standard set of buttons for editing. The Cut and Copy buttons cut and copy the current selection, respectively. The Paste button will paste the contents into the active window. Use the Undo button to reverse a previous action. Clicking the Redo button will redo the action.

3.12.3 ZOOM

The zoom options allow the user to control zoom operations in the model worksheet.

3.12.3.1 Zoom In

Zooms in on the center of the window.

3.12.3.2 Zoom Normal

Returns the zoomed in or zoomed out selection to normal size.

3.12.3.3 Zoom Out

Zooms out from the center of the window. It works opposite to Zoom In option.

3.12.3.4 Zoom to Fit

Zooms to fit the entire model in the window.

3.12.4 FIND

This command opens a different search window depending on where the search is initiated, i.e., the worksheet, the block's structure, or a database window.

3.13 WORKING WITH THE TEXT BOX

To add text to the model, select the Text tool (on the left in the Shapes toolbar), click on the model where the user wants to add text (or just double-click in the model window with Block/Text Cursor or All Objects Cursor) to start a text box; then type in the required text.

To stop entering text, click anywhere else in the model window. The user can access the text box after having stopped typing either by selecting the Text tool and clicking once on the text that has already been typed or by selecting the Block/Text or All layers tool and double-clicking on the text.

As the user types, the text box can be seen highlighted with a handle at the bottom-right corner of the text box.

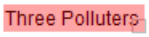


Figure 32: Text Box

To change the size and shape of the text box, click and drag on the handle. For example, to make the box narrower and taller, put the cursor over the handle. When the cursor changes shape to a two-headed arrow, drag it left and down.

Figure 33: Reshaping Text Box

Moving text around the model is easy. Before finishing the text entry, the user can move the text box using the mouse pointer and dragging it to the desired location.

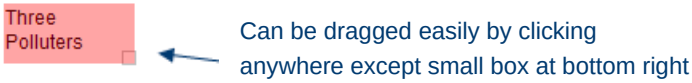


Figure 34: Moving Text Before Pressing Enter

To copy the entire contents of a text box to a model window, click once on the text box to select it and choose Copy from the toolbar or give the Copy command from the Edit menu. Then click on the model window where the user wants to place the text and give the Paste command. To copy all or a portion of the text in a text box and paste it into another text box or onto the model worksheet, double-click the text box (or click once using the Text tool) and select the text to copy. Then double-click to create or open a text box and paste the text.

The user can delete text by:

1. Selecting it with the cursor and pressing the Delete key
2. Choosing Clear from the Edit menu
3. Choosing Cut from the toolbar.

The user can also add formatting to text. To format existing text, click the text box to select the text, and then choose the formatting commands from the text toolbar (see Figure 33). The user can also change the color of the text.

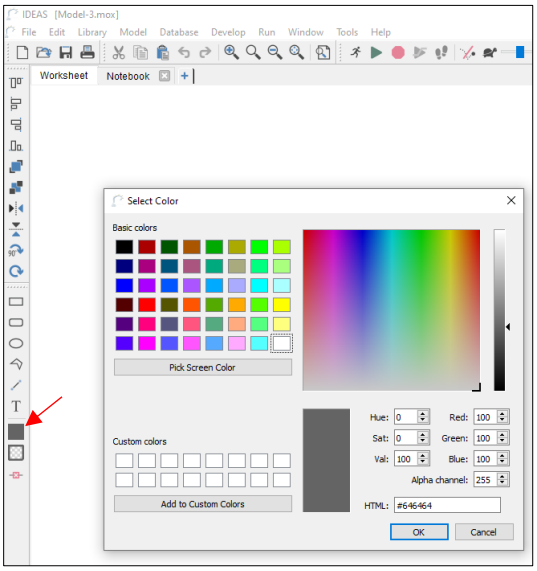


Figure 35: Text Tools and Color Palette

Note: If the user wants to type new text with a particular format, select the desired format before starting the text box. IDEAS will remember this format every time a new text is started. However, if the user changes the format of the text within an existing text box, IDEAS will not use that format for the subsequent new text. This rule, however, does not apply to font color. Once a font color is selected, it applies to all text written on the worksheet.

3.14 DRAG AND DROP EDITING

IDEAS supports a text editing feature called drag and drop. Drag and drop is the easiest way to move a selection of text a short distance or between documents. This feature is an alternative to the Cut, Copy, and Paste commands.

To move text using drag and drop editing:

Select the text to move. Point the selected text and hold down the mouse button. When the user drags the selection, an insertion point will appear to the left of the cursor. Drag the insertion point to the desired location and release the mouse button.

To copy text using drag and drop editing:

Select the text to copy. Hold down the Alt key, point to the selected text, and then hold down the mouse button while the user drags the insertion point to the new location. Release the mouse button.

Drag and drop can be used within a text box on a worksheet, a text file, an object's structure window and between any combination of the above with one exception. Drag and drop will not work between two separate text boxes. For example, if the user double-clicks on the worksheet to open a text box, enter text into the text box, then select a portion of the text, the user will not be able to move that text to another text box using drag and drop. This is because, on the model worksheet, there can be only one text box open for editing at a time. However, the user can drag the text to a text file window or an object's structure window.

3.15 DRAWING

Tools in the Shapes and Alignment toolbars are used to draw objects in the model and arrange them in layers. These tools are used to make the models easier to read or to make them more aesthetically pleasing.

The *Rectangle*, *Rounded Rectangle*, *Oval*, and *Polygon* tools add those shapes to the worksheet or Notebook. For example, to add a rectangle, select the *Rectangle* tool, click in the model where the user wants one of the rectangle's corners, and drag to the diagonally opposite corner. Similarly, the Line tools can be used to draw lines on the model. The Line tool draws a line at any angle and the *Draw right angle line* tool restricts the lines to horizontal and vertical lines. The user can also add colors to the shapes and lines drawn.

If the user holds down the Shift key as the *Rectangle*, *Rounded Rectangle*, or *Oval* shapes are re-shaped, the shape becomes a square or a circle. If the user resizes a square or circle while holding down the Shift key, it maintains its proportional measurements. This can also be achieved by right-clicking on the shape and selecting properties.

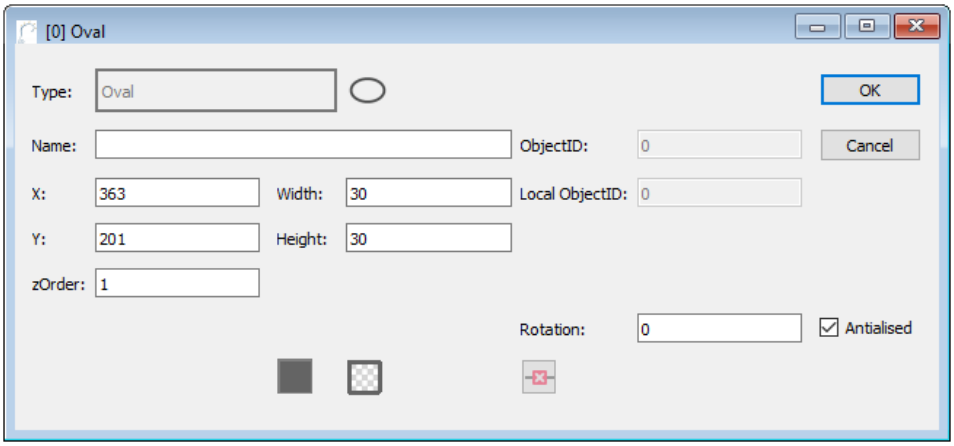


Figure 36: Adjusting Proportions of Shapes (Right-click > Properties)

The Shuffle Graphics tools let the user arrange drawing objects or text that are on top of each other. The user needs to select the object first and then select the *Bring to Front* or *Send to Back* button. The user can also make this selection from the dropdown from right clicking on the object.

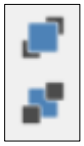


Figure 37: Shuffle Graphics Tools (Bring to Front and Send to Back)

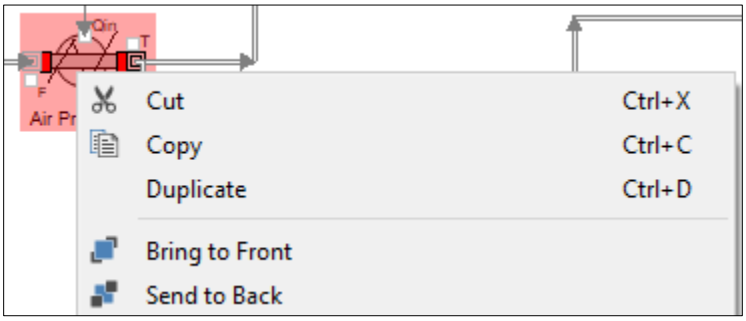


Figure 38: Shuffling Graphics by Right-clicking the Object

To see how the color and drawing tools work, select the Oval tool and draw an oval on the model. Note that, it comes out grey. Try changing the oval's color as well using the Properties menu, which appears upon right-clicking on the shape.

3.16 CLONING DIALOG BOX ITEMS ONTO THE WORKSHEET

This feature can be helpful in duplicating the dialog box information such as flow and temperature values onto the worksheet above the pipe or transmitter with which it is associated. Cloning allows the user to view this live data directly from the worksheet or notebook without having to open the dialog box. In very large models, having all the choices in dialog boxes can be a disadvantage. The user may want easy access to parameters in several objects that are scattered throughout the model. IDEAS takes care of this problem by giving the user the freedom to clone dialog box items and place them in a more convenient location.

The user can clone an item to more than one location. For example, the user might want a clone in two parts of the model window. Every clone acts exactly like the original: if the user changes the original or any clone, all instances are updated immediately.

Cloning dialog box items onto the worksheet provides direct control of the model. The user can use the dialog box items as the simulation runs, such as clicking buttons or entering values in text entry boxes. For instance, the user may have a model with objects that have controls in them that can be varied as the simulation is running. Instead of having those controls in a dialog box, the user can put them out on the model window. Alternatively, the user might want to have an area of the model that shows displays in a numeric form instead of the graphical form of the plotter.

To clone a dialog box item:

1. Open the desired dialog box.
2. Select the dialog clone tool.
3. Clone the variable window in the dialog box.
4. Click and place onto the appropriate spot on the worksheet.

Now, the user can watch the live data during the simulation without having to open the object dialog box.

To clone multiple items, select by dragging a frame over inputs or outputs you would like to clone. Then click the area you would like to place them on the model.

The user can then close the dialog box after cloning all the desired items.

Multiple copies of a clone can be made by cloning from the dialog box multiple times. Always make sure *Show Animation* is checked under the Run menu.

To change a value at a point in the simulation, click on the number in the cloned field and type a new value. A dashed highlight appears around the dialog field. Once the value is entered, the user must click outside the field for the new value to take effect.

The user can also clone from a dialog box within a hierarchical object to the hierarchical object's window or the main worksheet. To do this, open the hierarchical object's window. Open the object dialog box and choose the dialog clone tool, click on the desired item, and place it at the desired location.

Once a cloned item is on the worksheet, it can be moved anywhere. Many users prefer to have all the items together at one end of the model. (Notebooks are useful in such circumstance.) To move a cloned item, choose the dialog clone tool from the toolbar, click on the item, and place it on the worksheet/notebook.

Note: The graphs and data tables in plotters can also be cloned just like buttons and text entry boxes.

To size a cloned item, select the clone tool and click on the cloned item once. The user can pull at the handle to resize the cloned item. The user can also right-click on the clone dialog and change the dimensions via the Properties dialog that opens up. To identify the dialog box from which an item was cloned, choose the dialog clone tool and double-click on the item.

To remove cloned items from the model, simply select them (using the dialog clone tool) and press the Delete key or choose Clear from the Edit menu. If the original object from which a dialog box item was cloned is deleted from the worksheet, all clones of this item are automatically deleted.

3.17 LIBRARIES AND OBJECTS

Each object is represented by a unique icon that shows how that object relates to other objects in the model. Information comes into the object and is processed by the program that is in the object. Then the object transmits information out of it to the next object in the simulation.

Objects are stored in libraries. Because the worksheet only contains references to objects stored in libraries, the libraries must be present for the worksheet to run. There are many advantages to this method of using references to libraries instead of actual objects in models. If the definition of an object in a library is changed, all models that use that object are automatically updated. Also, object definitions are quite large, and storing just a reference to the library saves a great deal of disk space and memory (RAM). The entire definition for an object (its program, icon, dialog box, and so on) is stored in the library. IDEAS also stores the data that the user enters in an object's dialog box in the model.

There are two ways to open a library. IDEAS will automatically open all libraries associated with a model when the model is opened. This is controlled in the Options menu (under the Edit tab) by selecting the option *Automatic Search* in the Libraries tab. The user can also open a library by choosing *Open Library* from the Library menu. Once a library is opened, the library name appears alphabetically within the Library menu. User can see the list of libraires and objects in each library from the Library menu as well as drop any of the objects onto the worksheet from this menu. To see a list of objects in a library, select the library name in the Library menu. An alphabetical menu that contains a list of all the objects in the library will appear next to the library name. Selecting an object in this list allows the user to place the object on the worksheet.

Choose "Open" to locate and open another library where the object no longer exists in that location, or if another object cannot be substituted (as discussed below).

Choose "Select a Substitute Block" to select an alternative object from a library. A dialog box opens:

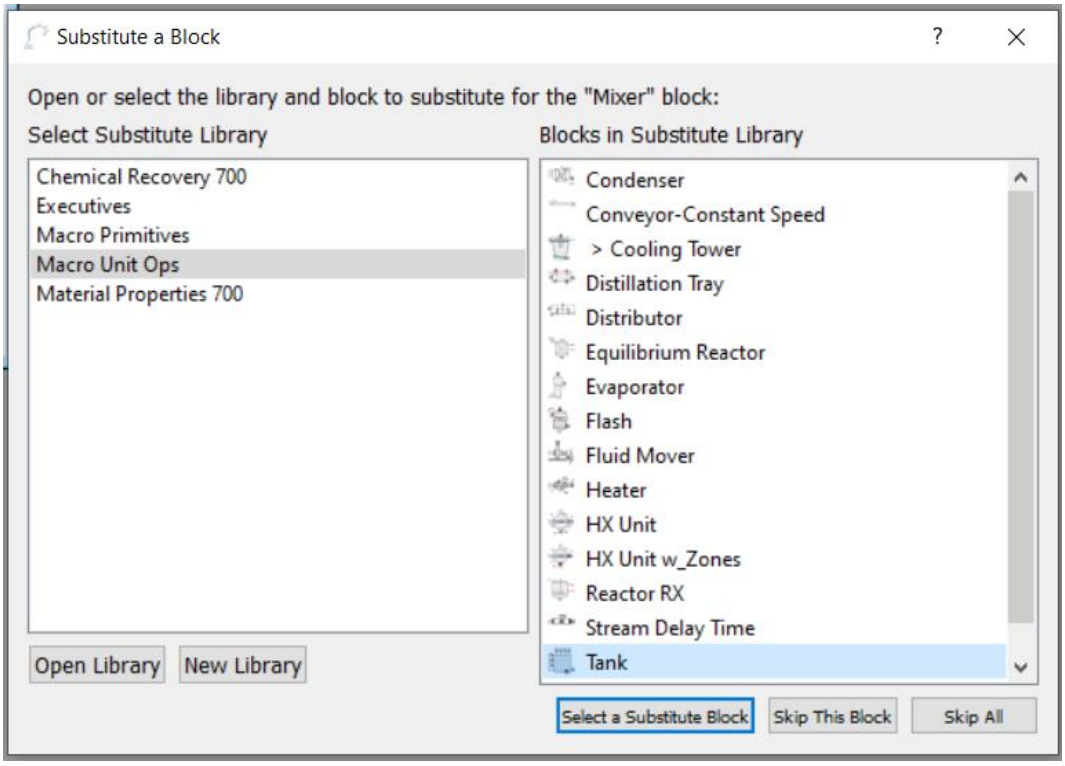


Figure 39: Substitute Object Dialog Box

Open libraries are listed on the left side of this dialog box; the objects in the selected open library are listed on the right. Use the dialog box to locate and select the object to substitute. The user can select an object from an open library, "Open" a library, or "Cancel" the search operation. Once an object is selected, choose "Substitute" and IDEAS will try to use that object in the model. (Any substituted object must be substantially the same as the object searched for, or IDEAS will report error messages). If the search is canceled, IDEAS will notify the user that it will place text in the place of the object (as mentioned above).

When the model file is subsequently saved, any new object name and/or location will be saved as well, so searching will not be necessary the next time the model file is loaded.

If an open library contains an object with the same name as the object searched for, IDEAS will attempt to use that object rather than search for the original library. When building new objects (hierarchical objects), do not use duplicate names for objects.

3.17.1 CHANGING LIBRARIES

If the "Automatic Library Search" option in the Options dialog box of the Edit menu is checked, IDEAS automatically opens the libraries that a model uses. However, sometimes the user may want to use objects from a different library (for example, when a new version of the library has been developed).

To bypass this process, unselect the "Automatic Library Search" option (Edit>Options). If this option is

unselected before opening a model, IDEAS will stop and request the location of each library used in the model. Then use the dialog box to find and open the desired libraries to be used in the model.

Once all of the required libraries are open, save the model. The next time the model is opened, it will automatically locate and open the libraries in the new location. Remember to reselect the "Automatic Library Search" option in the Options dialog box before reopening the model.

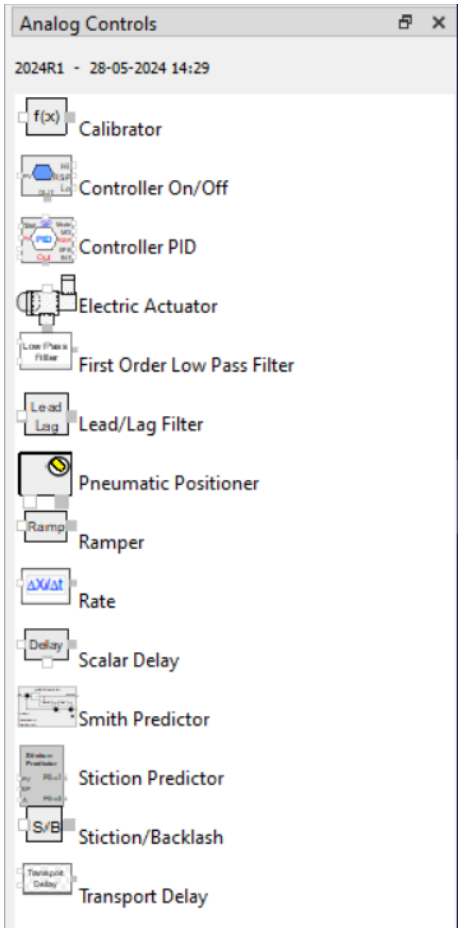


Figure 40: Library Window

3.17.2 LIBRARY WINDOWS

The first choice in the objects list for each library in the Library menu is Open Library Window. When this command is selected, IDEAS opens a library window for that library.

The top of the library window gives information about the library version and date last modified. The objects in the library are listed with pictures of their icons and their last modified date.

The primary method for opening a library window is by accessing the Library tab > Open Library Windows.

The library windows provide an alternative method for adding objects to models and help maintain the libraries (as discussed below). When using the Developers Kit to program objects, the library window can be used to access an object's structure.

To add an object to a model using the library window, click on the object in the library window and place it onto the worksheet.

To close the library window, click on the window's close box or choose the Close command from the File menu.

Note: This closes the library window but not the library.

3.17.3 MAINTAINING OBJECTS IN LIBRARIES

Note: This section is meant only for use on user-created hierarchical or Developer Kit objects, not on IDEAS core library objects. Changes made to these core objects may result in numerous errors. If any of these errors occur, call Simulation Support immediately.

IDEAS makes it easy to handle objects using the library windows. For example, to copy an object from one library to another, simply copy that object from its library window and paste it in destination library window.

To make a copy of an object in the same library, right-click on the object and select Duplicate Blocks. The Duplicate command copies the object into the current library. This process is common if the script in one object is needed as the template for a different object, or when two objects perform the same actions but need different icons.

To change the name of an object (NOT CORE LIBRARY OBJECTS), select the object in the library window, right click, and choose Rename Block from the menu. Then type in the new name. If the name of an object is changed, models that use that object will not be able to find it because they are expecting the original name. IDEAS then asks for object location and opens a search dialog box, as discussed in "Object searches." Use this search procedure to locate and substitute the renamed object. When the model is saved, IDEAS will save the new name so it will not prompt for the object when the model is opened again.

To remove an object from a library, select the object in the library window and press the Backspace key, Delete key, or right click and select Delete Blocks. IDEAS will not allow the removal of an object that is in use in an open model window. If an object used by a model is removed, and the model is opened later, IDEAS warns the user and puts a placeholder in the model window.

On very rare occasions, a message appears as a library is opened that indicates that an object has been corrupted or is bad. The corrupted object will appear in the library listing as "*BAD*Blockname." To save the rest of the library, copy the uncorrupted objects to a new library and discard the old library. Then copy the backup copy of the object into this new library. We encourage all users to frequently back up all their work within IDEAS.

3.17.4 ORGANIZING LIBRARIES

ANDRITZ strongly suggests that any new objects created (either hierarchical or with the Developers Kit) be stored in separate libraries created by the user. Do NOT put these objects into the Standard Libraries created by IDEAS. This arrangement would cause problems when updating libraries. Also, creating new libraries aids users in identifying and structuring their own developments. For example, worksheet-specific, user-specific, or project-specific libraries are good divisions. The user may want to copy core objects (those created by IDEAS) into these newly created libraries for easy access while building worksheets.

3.18 BUILDING A MODEL

This section describes the steps in modifying or creating a model from the libraries that come with IDEAS.

3.18.1 ADDING OBJECTS

The four steps to add objects to a model are:

1. Open the library, if necessary.
2. Place the object on the worksheet.
3. Move it to the desired position.
4. Connect it to objects before and after it in the model.

As stated earlier, the models do not actually contain objects. Instead, the model holds references to objects that are in libraries. In the model, the object's dialog box appears after double-clicking on the object's icon.

To copy an object into a model, the library in which that object resides must be open.

3.18.2 OPENING THE LIBRARIES

To open the library:

1. Choose Open Library from the Library menu.
2. In the File Open dialog box, locate and select the library.
3. Click Open to open the library.

3.18.3 PLACING AN OBJECT ON THE WORKSHEET

To place an object:

1. Click in the Library menu.
2. Drag down to the name of the library that holds the desired object. When the library is selected, the names of all the objects in the library appear in an alphabetical menu to the right.
3. Drag to the right and then down the list until the desired object is highlighted. Left-click the mouse button to select the object and then place the object at the desired location by again left-clicking on the worksheet.

3.18.4 MOVING OBJECTS

To move an object, click on the object and drag it with the mouse to the desired position.

3.18.5 CONNECTING OBJECTS

Connectors are used to hook objects in the model together. Connectors are linked by connections, the lines seen between the objects in the model window. For more information, see the Connectors and Connections section.

3.18.6 DELETING OBJECTS

Removing objects from a model is easy. Simply click on the object to select it and press the Delete key. Like other applications, selecting something and pressing Delete removes it from the worksheet. In this case, IDEAS removes the object and all input and output connections to the object. The objects can also be cut using the "Cut" command from the Edit menu.

3.18.7 CLOSING LIBRARIES

To close a library, choose Close Library from the Library menu and select the library to be closed.

Note: A library can be closed only when none of its objects are in use in the model.

3.18.8 DIALOG BOXES

Dialog boxes are pop-up windows for every object. They are accessed by double-clicking on the object's icon with the mouse. A dialog box is the user's interface to each object. It serves two main purposes:

1. Enter process or equipment information specific to the object to customize a generic valve, pump, tank, etc., in the library with specific conditions to run a particular simulation.
2. Display information about the object as the simulation runs either instantaneously or on demand. For example, a flow transmitter would display flow in the dialog box.

Click on the Help button at the bottom left-hand corner of the dialog box to get more information about the object. Click Cancel to close the dialog box and discard any changes made. Clicking OK or the close box at the upper left-hand corner of the dialog box will save the changes.

3.19 HIERARCHY

So far, each object described performs an action that is one part of the model. The action that the object performs is represented by the object's icon. Most IDEAS objects work this way. There is a second type of object—a hierarchical object—that works differently. A hierarchical object acts like a container that holds other objects. A hierarchical object may contain simple objects, other hierarchical objects, or both. The objects in a hierarchical object are connected just like other objects in a model, and hierarchical objects have input and output connectors like regular objects.

Although not needed to build models, there are many uses for hierarchical objects:

- Create advanced objects using a combination of simple, core objects.
- Simplify the worksheet by grouping areas of a complex model. These grouped objects can then be reused in other models without having to reproduce all the connections.
- Present a model as a few simple steps. To reveal the subsystems within a step, just double-click on the hierarchical object.
- Create many levels of hierarchy. Beginning with the simplest assumptions and proceeding to more complex ones allows the user to structure interconnecting processes and make models easier to follow.

3.19.1 OPENING HIERARCHICAL OBJECTS

As demonstrated below, the Heater/Boiling object from the Heat Exchangers library is a hierarchical object:

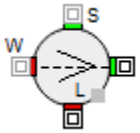


Figure 41: Heater/Boiling Hierarchical Object

It contains Temperature-Transmitter objects (Analog Controls library), a Pipe w/Heat Exchange object (Heat Exchangers library), and Tank/Multiphase (Tanks Dynamic library). When the user double-clicks on the object, the following window opens:

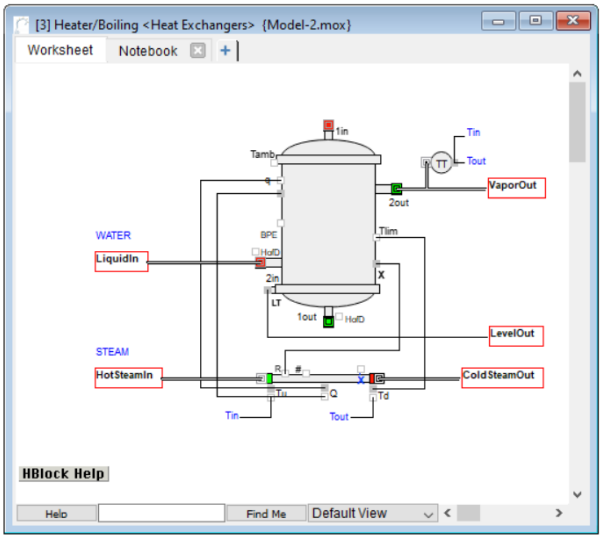


Figure 42: Heater/Boiling H-Block

3.19.2 INTERNAL STRUCTURE OF HEATER/BOILING OBJECT

It should be noted that the hierarchical object itself is only a picture representation of this window. Connections seen here are linked to other objects on the main worksheet. These connections are represented by text boxes (named connections with borders around them).

Changes can be made in these objects just as in objects on the model worksheet. Open the hierarchical object and double-click on the desired icon to change its settings. Dialog box items from these objects can also be cloned as discussed previously in the Cloning section.

3.19.3 CREATING A HIERARCHICAL OBJECT

Please see Appendix C for detailed instructions on how to create hierarchical objects.

3.20 PLOTTERS

Most of the IDEAS plotters have many features in common. This section describes IDEAS plotter dialog boxes and how to customize plots.

Up to four (4) traces can be plotted at a time. Plotters can only be connected to scalar connectors (refer to the Connectors and Connections section. Therefore, a transmitter or another object with scalar outputs needs to be on the worksheet when using a plotter. More than one plotter can be used on a worksheet, and they can be placed at any point on the model. The choice of plotter types depends on how the user wants information plotted. See Plotters help text for details on different plotter types and the plotter toolbar.

When a user clicks on a plotter icon, the trend (Graph tab) is opened. A plotter window has three (3) tabs. The upper tab shows the Dialog, the middle tab shows the Graph, and the third tab shows the Data for that plot. Most of the time, the user is interested in seeing the plot; however, the data tab is also quite valuable.

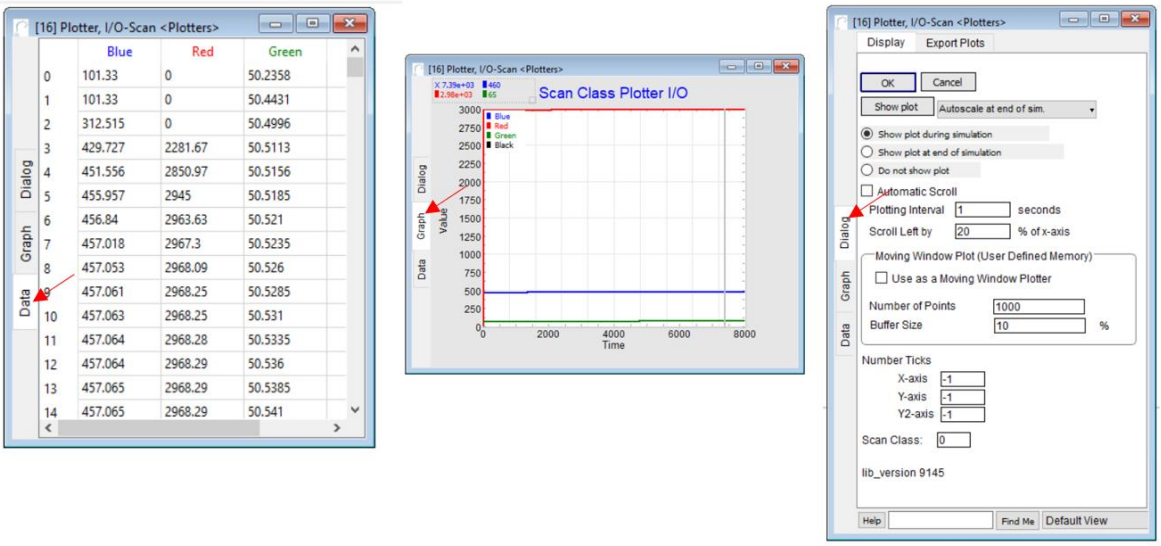


Figure 43: Plotter I/O Window

3.20.1 PLOT TAB

To change the plot labels and axis limits for the plot, right-click on the plot pane and choose Graph Properties option (see Figure 44).

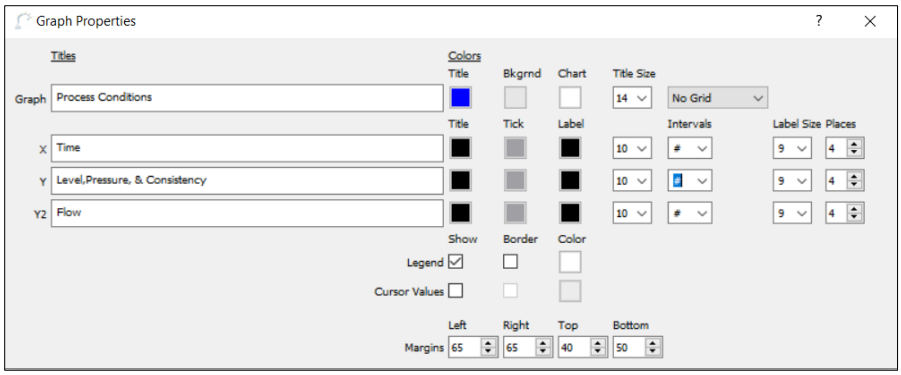


Figure 44: Graph Properties

3.20.2 DATA TAB

When the data in the data pane is changed, those changes are reflected in the plot. Numbers can be changed; rows can be inserted by pasting, and so on. Use this capability to view how various data would be plotted or to plot a reference line.

3.20.3 PLOTTER TOOLS

Note the trace editor tools can be accessed by right-clicking on the plot and selecting from the options (see Figure 45).

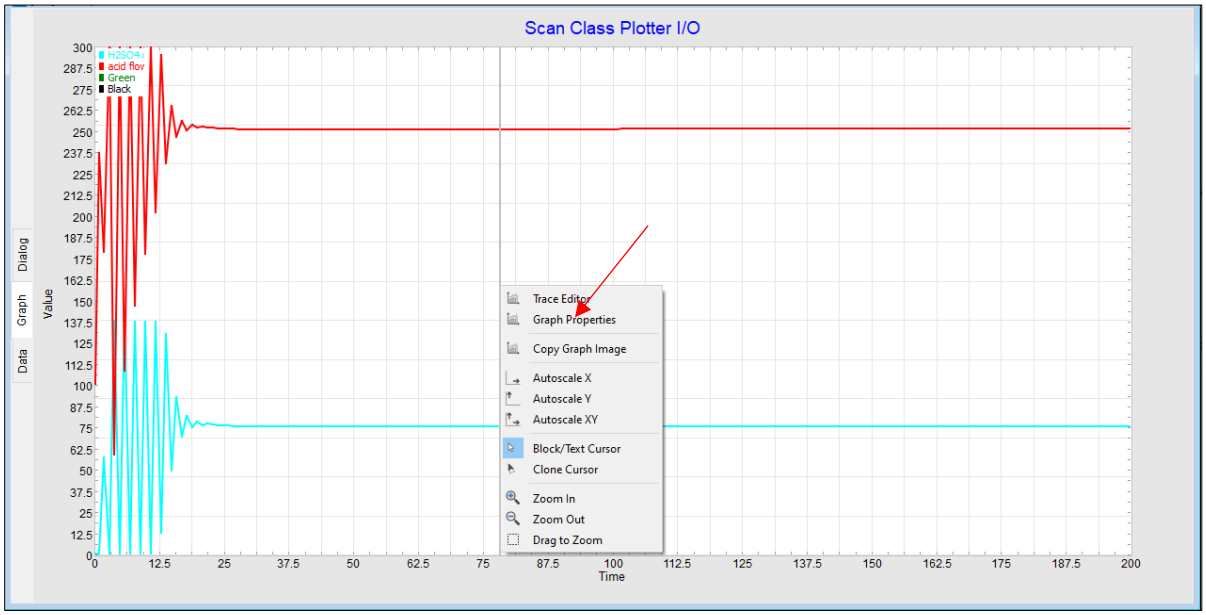


Figure 45: Trace Editor Tools

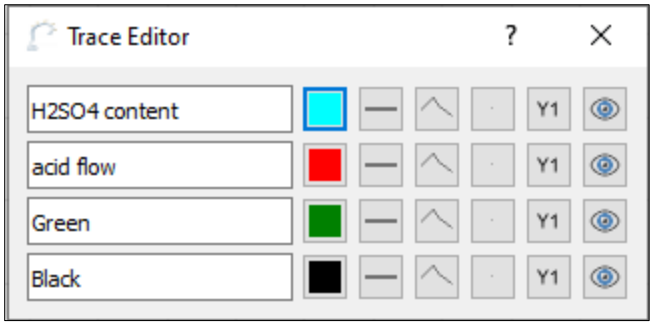


Figure 46: Trace Editor

For complete information on the functions of each tool, see the Plotters help text.

3.21 CONNECTORS AND CONNECTIONS

The small squares attached to each side of the object are called connectors. Information flows into an object at input connectors and out of the object at output connectors. An object might have multiple input and/or output connectors, and some objects have none. Connecting lines are used to link objects together. These lines (called connections) show the flow of information from object to object through the model. Major equipment objects are typically connected with piping objects. When these connections are made with the mouse, all subroutine links are made and parameters passed to the downstream object. In fact, flowing through the pipe is an array of components with their associated properties and process conditions.

There are several types of connectors in IDEAS. Many objects use value (scalar) input and output connectors to pass values:

Figure 47: Scalar Input and Output Connectors

These connectors pass single values and are mostly used for control signals, transmitter outputs, etc.

The actual process streams (flow through pipes), which consist of arrays of numbers, use stream array input and output connectors:

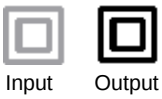


Figure 48: Stream Array Input and Output Connectors

The method of passing these arrays is part of the IDEAS proprietary methodology, and therefore non-IDEAS objects will not successfully communicate through these connectors. All objects using the scalar connectors will directly connect to scalar connectors on IDEAS or any other object using scalar connectors (Stream array objects can be created using IDEAS Developers Kit).

IDEAS ensures that correct connections are made. For example, if the user tries to connect a stream array output to a scalar input, IDEAS will provide an error dialog message.

To connect an output connector on object A to the input connector on object B:

Move the cursor to the output connector of object A. The cursor changes from an arrow to another cursor with arrows at both ends.

Click the mouse button at the output connector, then drop the line to the input connector on object B. The line becomes thicker when the double arrow is over the connector.

A dashed line indicates an incomplete connection. To erase the faulty connection, click on the line so that it thickens (indicating that it has been selected). Press the Delete or Backspace key.

3.22 CHANGING CONNECTION CHARACTERISTICS

The lines that make up connections can be formatted to make the model layout clearer. The Connection Lines command from the Model menu changes the line style of the connections. To use the command, select the desired connections and choose an option from the hierarchical menu:

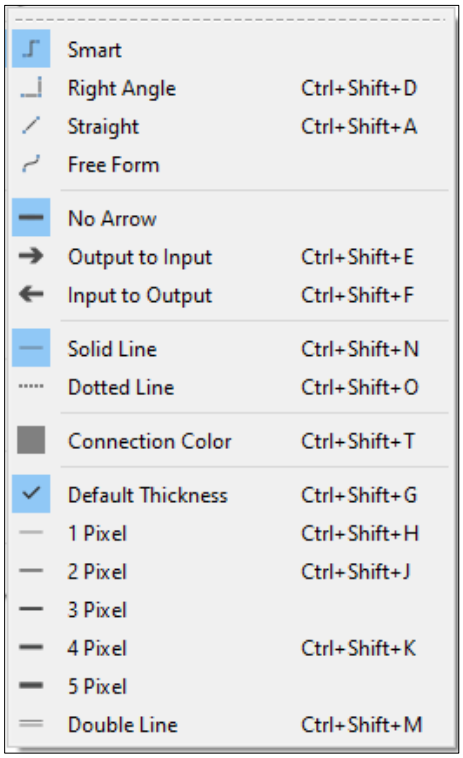


Figure 49: Connection Line Styles

3.22.1 CONNECTION LINES COMMAND

The top option allows the user to choose Smart, Right-Angle, Straight, or Free Form connections. The right-angle option will make the selected connection into a three-part connection with anchor points. The default type of line can be specified in the Options command from the Edit menu.

The second connection line choice provides arrows on the connections. Arrows can be used for all connections, and the direction of the arrowhead follows the direction the line was drawn when the connection was made.

The connection lines can be thin, medium, thick or double lines based on the option selected. IDEAS normally uses thin lines for clarity, but thick lines may be used to highlight important connections. The fourth choice allows right angle connections appear as solid lines or dotted lines. Connection colors can be changed through right-clicking the connection and choosing the color.

3.22.2 WAYS OF CONNECTING OBJECTS

There are five ways to make connections between objects (four as shown in Figure 50 below and connection via text block):

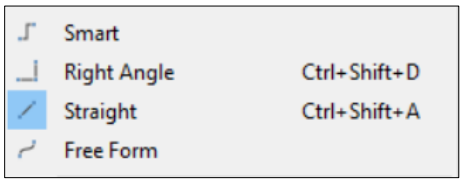


Figure 50: Connection Options

Use a Right Angle connector as shown below:

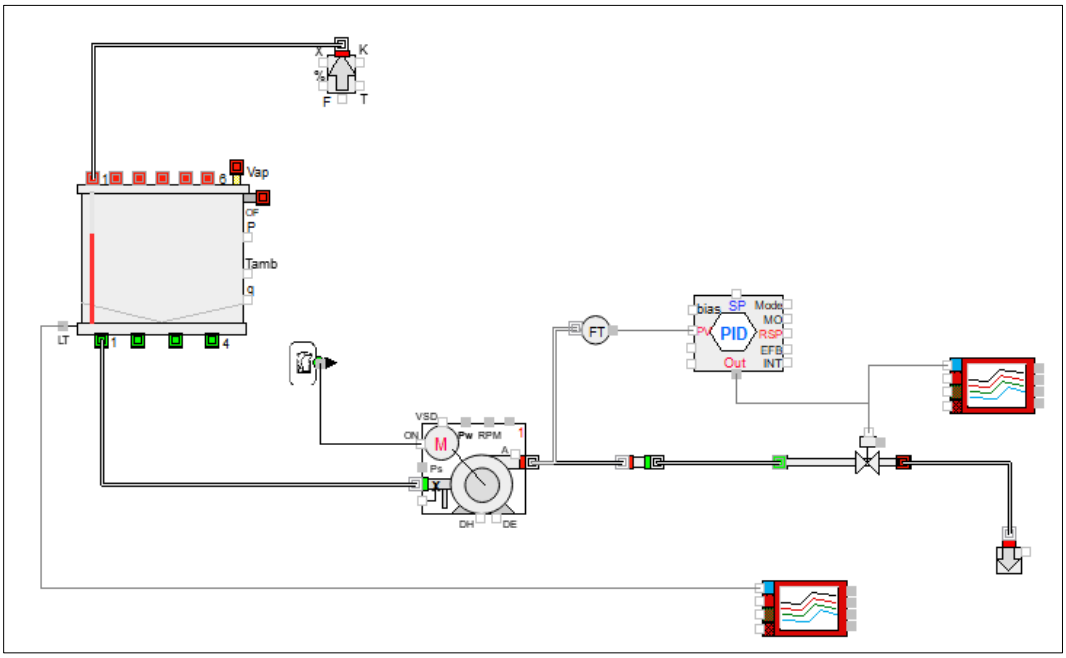


Figure 51: Right Angle Connection

Use a Straight connector as shown below. Draw a connection from the connector to a point below it, then release the mouse button. Make sure that the cursor still looks like a double arrow; immediately click again and drag to the desired connector. Release the mouse button when the line thickens. The result will look something like:

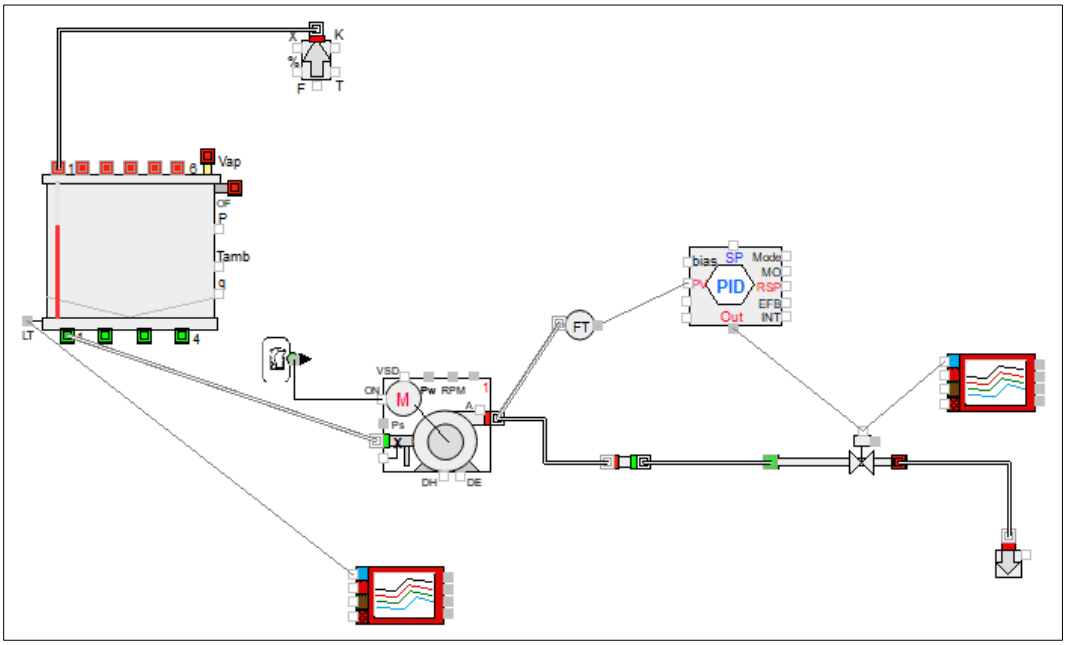


Figure 52: Connection Through an Anchor Point

Use a named connection. Named connections are text labels that are used to represent transparent connections in the model. Double-click near the connector to obtain a text box. Type a name for the connection. Repeat this procedure at the desired destination connector using the same label (or simply copy and paste the text box). In named connections, the user must use identical spelling in the text names (including spaces). Connect the text box with the technical pen tool to the corresponding connector. To see the connection, choose Show Named Connections from the Model menu.

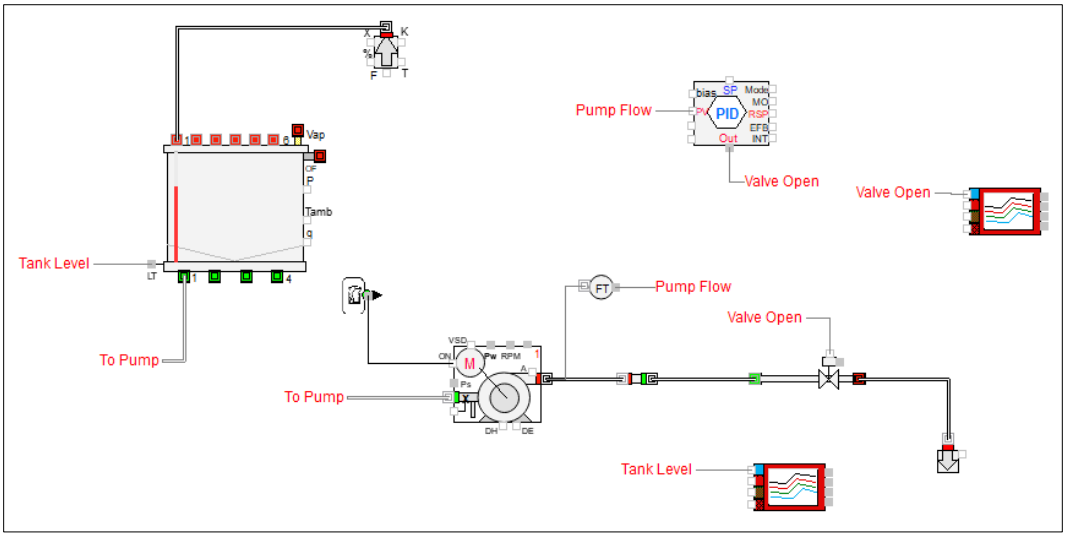


Figure 53: Connection Using Named Connection

3.23 RUNNING THE MODEL

3.23.1 SIMULATION SETUP

The Simulation Setup dialog box specifies how the simulation will run and for how long. The dialog box looks like:

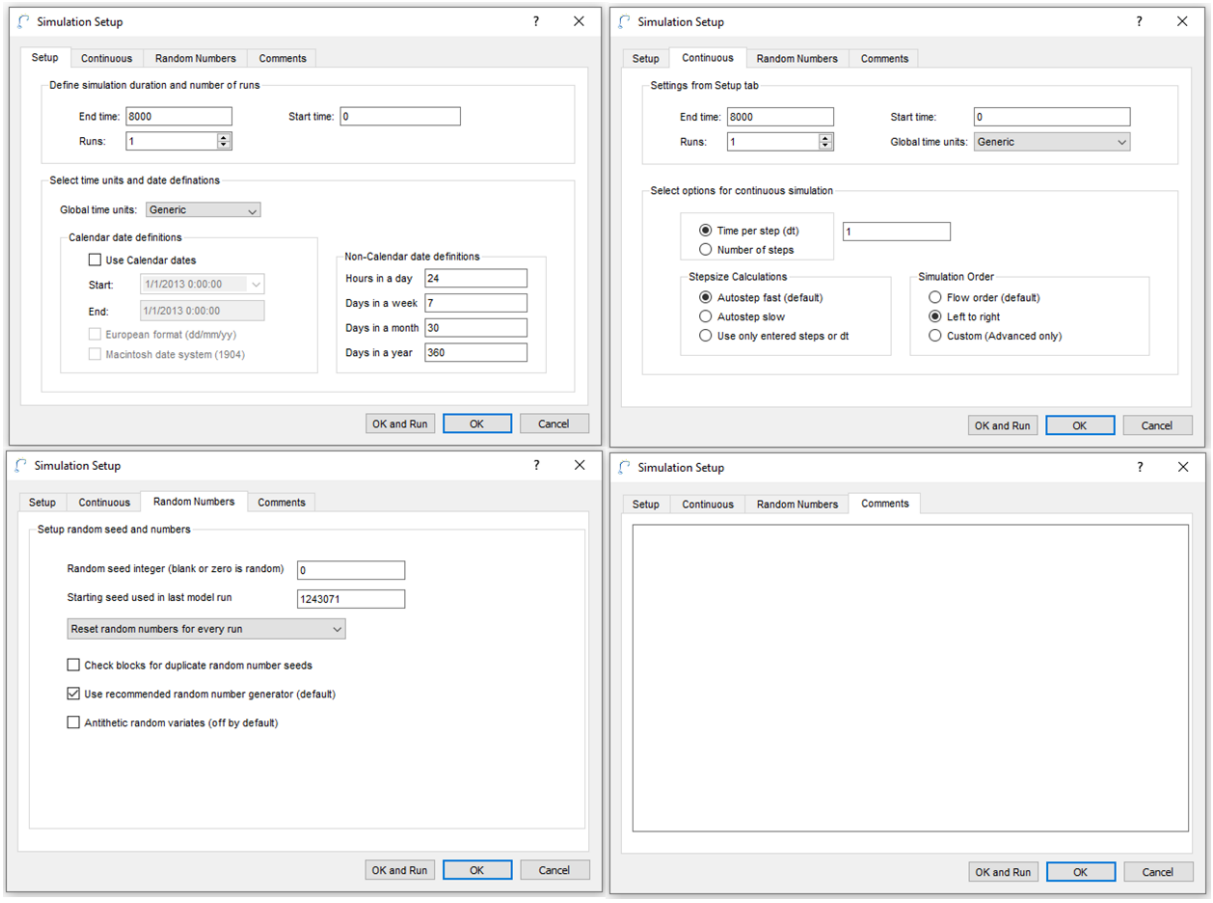


Figure 54: Setting Up Simulation

3.23.2 SIMULATION SETUP DIALOG BOX

Note: Always select Left to Right for Simulation Order when running dynamic simulations. In steady state (Macro) models, switching to the Flow Order may improve the speed at which convergence is reached. The user may want to experiment with the simulation order for a given Macro worksheet.

Each time a simulation is run, IDEAS uses the same values entered in the Simulation Setup dialog box from the Run menu.

Generally, the only setting to be changed in the dialog box is the end time. For most purposes, the

simulation should start at the beginning, so the default start time would be 0. Also, most continuous simulations use a Time per step (dt) of 1. Change the Number of runs option to repeat the simulation and to look at how results change over many runs.

3.23.3 SIMULATION STATUS

The choices in the Status bar at the bottom of the screen are based on the settings in the Simulation Setup dialog box.

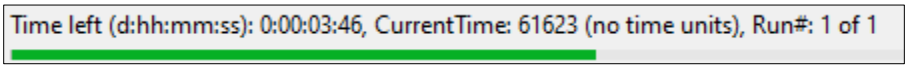


Figure 55: IDEAS Status Bar

The numbers after the Time left are an estimate of the time remaining in the simulation, so the user can determine how long it will run. The CurrentTime shows the current time of the simulation in time steps. "Run" is the number of the simulation, if running multiple simulations. These values are determined by the entries in the Simulation Setup dialog box.

Note: The Time left shown in the status bar is only an estimate.

3.24 SENSITIVITY ANALYSIS

Sensitivity analysis allows the user to conduct experiments and investigate the effects of changes in a structured, controlled manner. The user does this by running simulations many times, changing the value of a variable or numeric parameter each time the simulation is run. When the user repeatedly runs a simulation that has parameters that vary with each run, the user can see the range of the results and look for trends or anomalies. For example, if the user runs the same simulation a hundred times, the user can look for extremes in results and look for averages of critical values.

IDEAS' sensitivity analysis feature gives the user the ability to explicitly specify individual parameters to change and provides several methods for changing them.

Sensitivity analysis differs from using random numbers as inputs in the model. When the user runs a model and uses a random input, a random number will be input at each step or event. When the user uses sensitivity analysis, the randomness occurs from run to run, not between steps in the run. This feature is particularly useful when the user wants to investigate how a change in one parameter impacts the pattern of behavior for the entire model.

To use sensitivity analysis in a model:

- Choose a parameter value to investigate
- Choose how the parameter will change
- Select the number of runs for the analysis
- Run the simulation with Use Sensitivity Analysis selected in the Run menu

Sensitivity analysis is a very powerful feature that the user can use in models. For complete details, see Appendix D, Sensitivity Analysis.

3.25 MONITORING MODEL OPERATION

The following sections describe various methods for monitoring the operation of the model using the IDEAS simulation package. The first five methods described are available to the user at the IDEAS workstation, while the sixth method described is only available when the model is used for training.

3.25.1 PLOTTERS

The user can choose to show the plot during simulation, at the end of simulation, or not show the plot, which can be selected via radio buttons from each plotter's dialog box (see Figure 56). If the user chooses to show plot during simulation, the plotter will automatically open the plotter object to the graph tab. Information is Continuously Updated on the plot, allowing the user to visually monitor the progress and possible errors in the model.

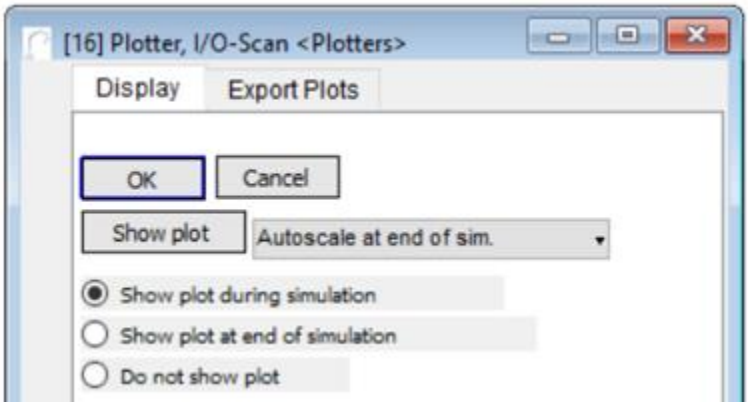


Figure 56: Options for Plot Display

3.25.2 ANIMATION

For additional feedback as the model runs, IDEAS has animation capabilities. Many of the objects in the libraries have animation built into them, such as fluid levels in tanks or color indicators of energy flow. The animation in these objects is described in the library manuals. Also, the help text in the dialog box for an object tells whether it is animated and, if so, which aspects are animated.

To see animation, choose Show Animation from the Run menu so that the command is checked, then run the simulation. Animation is not shown all the time as it increases simulation time. Animation may be left on while changing, debugging, or presenting the model and then turned off when running the model for analytical results. Selecting bitmap objects in the Options dialog box speeds up animation and the redraw of objects.

To see the animation without interference from the plotter, plotters can be configured to remain closed during simulation runs. (The plotter can be prevented from opening at all during the simulation by choosing the dialog tool in the plotter's tool pane and deselecting Show plotter during simulation. This option can be completed manually after the simulation starts by clicking the plotter's close button).

The Show Object Messages command from the Run menu is a type of animation used by programmers when they debug objects that they build.

Note: Animation used extensively will slow down model execution significantly.

Figure 57: Animation Tool Bar

In IDEAS, users can choose the animation speed from the Animation Tool Bar. Dragging the slider to the extreme left will slow down the animation speed and bringing it to the extreme right enables maximum speed of animation. User can also choose to run animations at an intermediate speed, but it is recommended to use the fastest option, which is set as default.

3.25.3 DIALOG BOXES

Dialog boxes can remain open while the model runs, although it slows down the simulation significantly. Settings in a dialog box can be changed while the simulation runs such as choosing different radio buttons or typing in new values. When a button is clicked, the object reflects that change on the next step. However, when text or numbers are entered into a dialog box field, the field gets highlighted. The modified value takes effect only when the user comes out of the field by clicking elsewhere on the dialog box.

3.25.4 CLONING DIALOG BOX ITEMS

Cloning allows the user to view live data directly from the worksheet or notebook without having to open the dialog box.

Note: This feature allows the simulation to run faster than open dialog boxes. Cloning dialog box items onto the worksheet provides direct control of the model. The user can use the dialog box items as the simulation runs, such as clicking buttons or entering values in text entry boxes.

To clone a dialog box item:

- Open the desired dialog box.
- Select the dialog clone tool.
- Clone the variable window in the dialog box.
- Click and drop onto the appropriate spot on the worksheet.

Now, the user can watch the live data during the simulation without having to open the object dialog box. For multiple items, select by dragging a frame or holding down the Shift key when selecting more items.

Then close the dialog box by clicking the close box or choosing the Close command from the File menu.

The user can make multiple copies of a clone by cloning from the dialog box again or making a clone of another clone. Always make sure Show Animation is checked under the Run menu. For additional information on cloning, see the Cloning section earlier in this section.

3.25.5 USING THE NOTEBOOK

Each model can have multiple Notebooks that can be used for controlling the model parameters, reporting simulation results, and documenting the model. This Notebook feature provides more flexibility with cloned dialog box items. To have easy access to cloned items, they can be collected in a model's Notebook, producing a single data monitoring area. The user can even rename notebooks for better representation of area in a particular model.

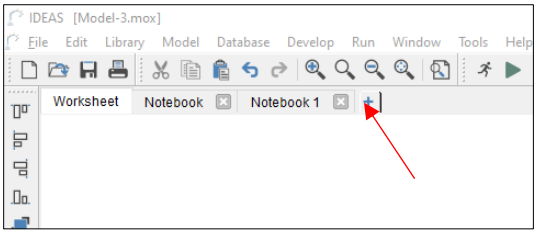


Figure 58: Adding a Notebook

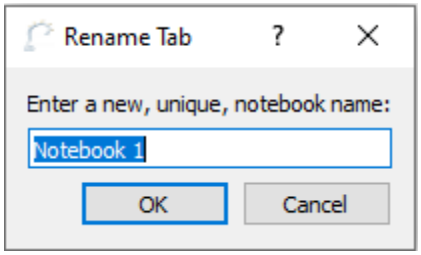


Figure 59: Adding and Renaming Notebooks

Use the Notebook for both input and output values. For example, the user may want to clone buttons in the Notebook and change them as the simulation progresses, or the user might want to use the Notebook mostly for looking at the various outputs of the model. Plots can even be cloned into the Notebook. Another common use is as a control panel. The Notebook is also handy for combining all the outputs into one spot for printing or making a screenshot for a report.

3.25.6 DCS MONITORING

IDEAS has the ability, through a custom interface, to communicate with a DCS. The model can be set up to simulate a mill or plant by responding to outputs from the DCS with inputs that are both analog and digital. The simulation computer controls the model simulation, transparent to the operator, just as the configuration works transparent to the operator in the actual plant. Once the simulation has been started,

the DCS console will be the only interface necessary. The operator will use the DCS console to interface with the system as if actually running the plant from the control room. The operator can page through the graphics and view the conditions of the process, change set points on the controllers to monitor the results, and start/stop motors.

SECTION 4. SPECIAL FEATURES

4.1 STREAM NUMBERING

The user can name the IDEAS streams using the N-Click feature. To make this work, the worksheet must contain the Stream Number Center object from Executives library.

Open Stream Number Center dialog box. In the Inputs tab, select SN Tag type as Consecutive Numbers/From BN-To BN and type S: in the SN Tag prefix field. Now, the user is able to name the stream with N-Click. Hold N key and left-click with mouse on the connection line. The user can hide all the Stream Number objects with the Hide SN objects option checked. See Figure 60 below for details. The syntax of stream numbers follow with S :< BN1>-<BN2>. The user can see all the Stream Numbers in the table provided under the Displays tab.

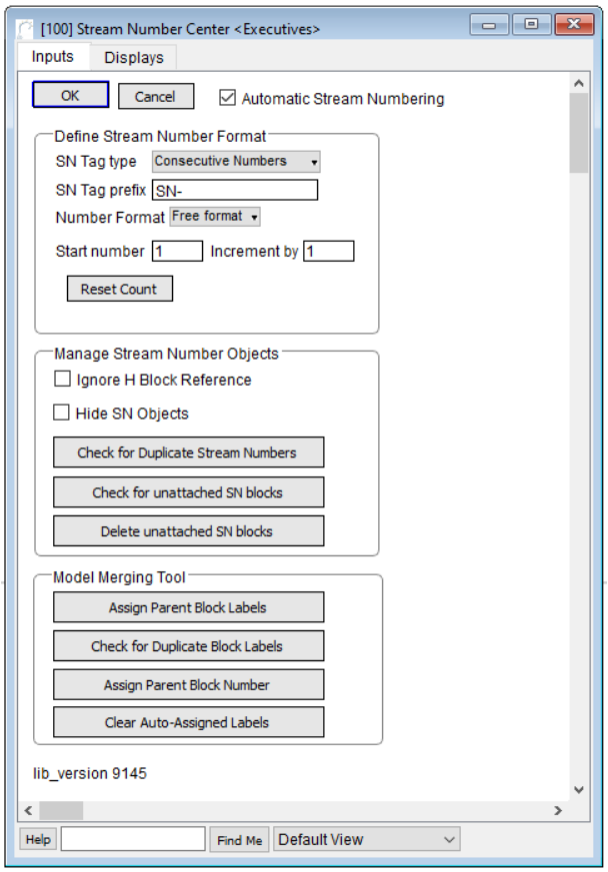
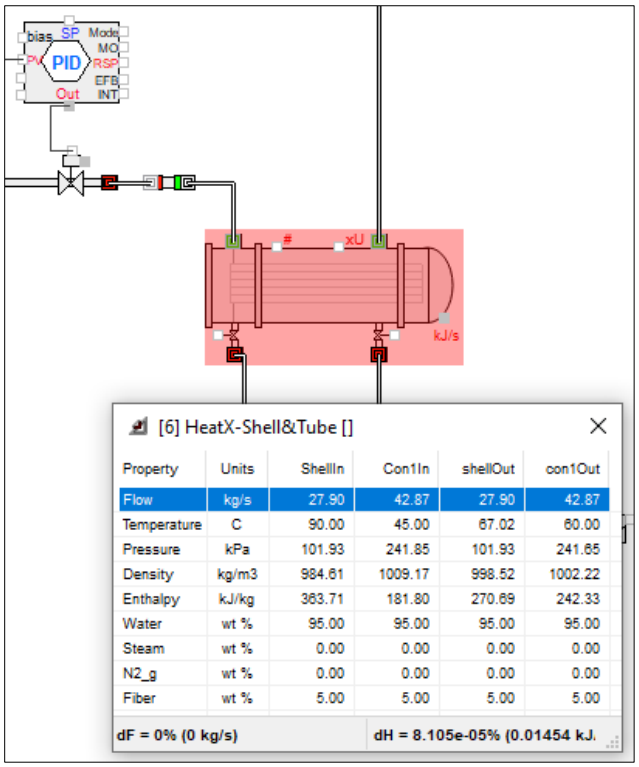


Figure 60: IDEAS Stream Number Center Dialog Box

4.2 S-CLICK

The user can see the simplified details of an object or stream by S-clicking on the object or stream. This feature will only work when the simulation has been run at least once. Hold down the S key on the keyboard and click on an object with the mouse. A table will appear that shows information such as flow into and out of an object and various other data. Please note that this feature will not work for Hierarchical objects. See Figure 61 below which is an S-click display on a heat exchanger object.



Property	Units	ShellIn	Con1In	shellOut	con1Out
Flow	kg/s	27.90	42.87	27.90	42.87
Temperature	C	90.00	45.00	67.02	60.00
Pressure	kPa	101.93	241.85	101.93	241.65
Density	kg/m3	984.61	1009.17	998.52	1002.22
Enthalpy	kJ/kg	363.71	181.80	270.69	242.33
Water	wt %	95.00	95.00	95.00	95.00
Steam	wt %	0.00	0.00	0.00	0.00
N2_g	wt %	0.00	0.00	0.00	0.00
Fiber	wt %	5.00	5.00	5.00	5.00

Figure 61: IDEAS S-Click Display

4.3 D-CLICK

Another way to monitor objects is by using the D-Click feature. For this feature to work, the Display Manager object from Executives library should be present on the worksheet. Hold down the D key on the keyboard and click on an object with the mouse (the simulation does not have to be running). A display will appear that shows information such as flow into and out of an object and various other data (see Figure 62 below, which is a D-click on the Heat Exchanger object).

The user can also change the display parameters and units for both the S-click and D-click displays using Display Manager Object. Go through the help file of the Executives library to know more about Display Manager Object.

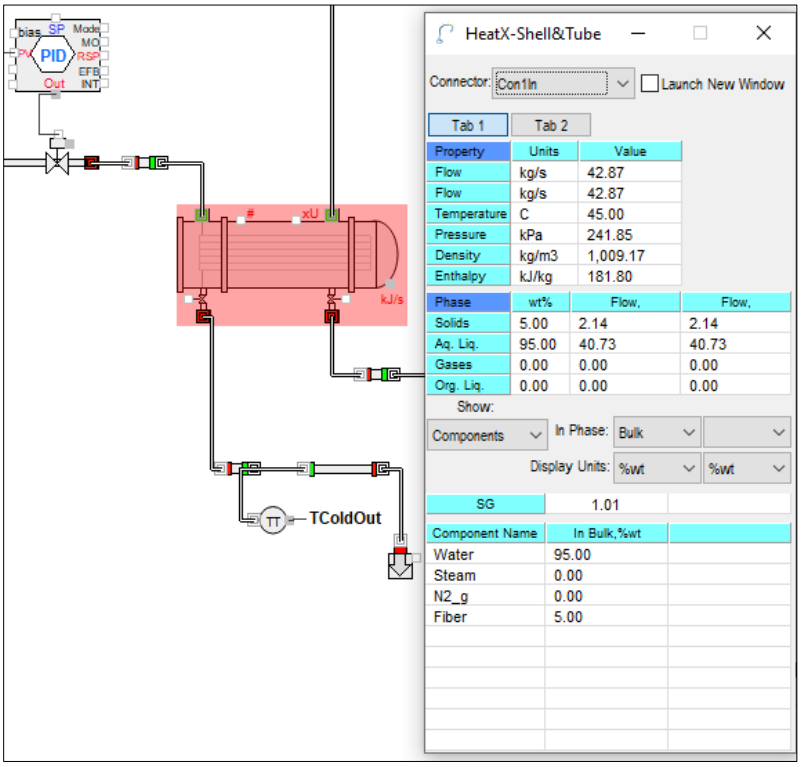


Figure 62: IDEAS D-Click Display

4.4 SCENARIO IMPORTING

The user can send parameters to IDEAS model from outside IDEAS using Scenario Importer object which is under Dynamic Data Exchange library. For more information, please refer the IDEAS help text of the Dynamic Data Exchange library.

4.5 STREAM EXPORTING

The user can send parameters and stream information of IDEAS model to outside IDEAS using the Stream Exporter object which is under Dynamic Data Exchange library. For more information, please refer to the IDEAS help text of the Dynamic Data Exchange library.

4.6 DYNAMIC DATA EXCHANGE

IDEAS is capable of sharing data with another application by creating a DDE link. For more information, please refer the IDEAS help text for the Dynamic Data Exchange library.

SECTION 5. CONTACT US

5.1 IDEAS PRODUCT TECHNICAL SUPPORT

The user can reach product technical support by calling one of the numbers listed below. Technical Support is available for all registered users.

- Support Email: simulation.support@andritz.com
- Sales E-mail: automation-sales@andritz.com
- Website: www.andritz.com/automation.htm
- Phone: (USA) +1 770 640 2500; EST 8:00 A.M. to 5:00 P.M.

General comments regarding IDEAS Simulation Support:

- Software support inquiries should be directed to our support team email: simulation.support@andritz.com or call us at the phone number shown above. Using the email address ensures that someone will receive your inquiry even when others are not in the office.
- Please send us the model in question, or a representative smaller model, the IDEAS Version Number, error message log, any custom libraries, and any applicable snapshot of the issue if applicable so that we may resolve the issue as quickly as possible.
- We also may require the IDEAS Serial Number.

5.2 REMOTE DIAGNOSTIC

Remote diagnostic support may be a possibility as well.

5.3 IDEAS AUTHORIZED TRAINING SESSIONS

IDEAS provides Hands-On training courses to assist users in learning how to utilize our powerful dynamic simulation tool. We provide standard 3-, 4-, and 5-day training sessions, and also customer training sessions.

Appendix A. Pressure/Flow Network

The IDEAS pressure/flow network solution method is unique in many ways. The connection of the various objects together is all that is necessary to create the global solution array that the global solver converges.

The global pressure/flow solution is quite robust and stable with most piping configurations. It is possible, however, to create configurations that will have trouble converging. In the incompressible network, the solver generally has trouble converging a node that is connected to a pipe that has a very high flow to pressure relationship (i.e., a very small delta pressure causes a very large flow). This translates into very large diameter or very short pipes. When experiencing stability problems in a node (i.e., the pressure is oscillating and the flows into and out of the node are not balanced), look at the pipes that are connected. If one or more are very short or are very large in diameter then that is the most likely problem. Depending upon the situation, you may be able to combine several pipes into one longer one or remove the short pipe if it is negligible to the solution. Alternatively, the user may be able to increase the pressure drop in the pipes (effectively making them longer) by artificially increasing the fitting resistances or pipe roughness values if the exact pressure drop calculation in the short pipe is not critical. Finally, if the exact piping configuration is necessary and it is unstable, then the delta time can be reduced to increase stability. This reduction will consume more CPU time, but should help the unusual pipe configuration to converge. The Solver-Fluid Flow object dialog box will reveal the block number of the node with the maximum flow error and the magnitude of the convergence error at that node. The Find Block function under the Edit menu will go to the offending node.

The solver has global access to all of the stream arrays on the worksheet as well as the critical object parameters like pipe dimensions, fluid viscosities, etc. At the end of each simulation step, the solver provides an emulated simultaneous convergence. The solver sequentially accesses all pressure/flow objects on the worksheet; then after grouping them to minimize execution time, the solver proceeds through each pressure/flow object converging the nodes and calculating the flows for the flow objects until the user-specified convergence is reached. The equations used are given in the specific documentation for the pipes/valves, nodes, and solver objects.

The hybrid solution techniques (combining sequential modular, emulated simultaneous, and simultaneous techniques) used allow the user to quickly isolate any problem areas on the model worksheet. Many built-in user messages lead the user through any possible interconnection problems.

A.1 Incompressible and Compressible Fluids Capability

There are two different pressure/flow networks: one for incompressible fluids and one for compressible fluids. They both use the same objects and are selected in the dialog boxes of the pipes, valves, and pressure nodes.

The compressible solution technique is more demanding than the incompressible technique. The IDEAS compressible solution is completely implicit and uses a matrix technique to solve for node pressures. The compressible solution technique takes into account the momentum balance, which allows for modeling the "mass" or "pipe packing" phenomenon in gas pipelines. The incompressible network is a lower fidelity solution technique, but it is faster in terms of execution speed and is adequate in fidelity for almost all incompressible flow applications.

When using the compressible option, a smaller delta time step is often required to achieve numerical stability and convergence, thereby, consuming more CPU time. The delta time can therefore be set independently for the compressible and the incompressible portions of a model in the Discrete/Continuous Executive object (see the description of scan classes in Appendix B).

A.2 Pressure/Flow and Boundary Objects

Objects whose flows and pressures are calculated by the global solver are often referred to as Pressure/Flow objects. These include pipes, valves, pumps, compressors and pressure nodes as well as some other more complex equipment that, due to the fast response nature of its flows or pressures, are included in the global solution. Objects that calculate their own flows or pressures independent of the global solver are objects that by their very nature respond slowly enough that they appear to be external sources or sinks to the pressure/flow network and are referred to as Boundary objects. Boundary objects solve their internal equations via sequential or simultaneous solution methods as selected by the object developer.

Certain rules describe the way objects must be connected on the worksheet for the pressure/flow solver to function properly. First, the user will see two different types of connectors on objects; a single box (an input connector looks like this and an output connector looks like this); these connections are scalar signals or just a single number. These connections are used for control signals and other data paths that require only a single number. The other type of connection is a double box (the input connector looks like this  and the output connector looks like this  . These represent stream array connections. Stream connections pass an array of information through them with information such as temperature, pressure, flow, enthalpy, density and mass fractions of various components in the stream. Stream connections are used wherever a process stream passes. Some objects can have both types, for example a flow transmitter has a stream array input connection, which connects to another stream array connection and a scalar output connector which provides a flow value to a controller, plotter, or other object.

All objects that have stream connections are color-coded and by connecting green to green and red to red the rules are automatically followed.

Note: Even though some green to red stream array connections can be made on the worksheet, the simulation will not run.

A simple explanation of the purpose of these color-coded connections follows:

When two objects are connected together with stream connections, the connection represents only one specific place in the piping system as shown below.

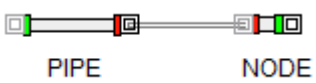


Figure 63: Color-coded Connections

(Connect GREEN to GREEN and RED to RED)

Only one array is created in computer memory to which both connectors, and therefore both objects, have access. Since there is only one place to put the flow value and only one place to put the pressure value for this point, both objects cannot define both values or they would overwrite each other. There are, therefore, two different types of stream connections:

FLOW CONNECTION: The object with a FLOW connection can define the flow at this connection but not the pressure. A flow input connection is GREEN. A flow output connection is RED.

PRESSURE CONNECTION: The object with a PRESSURE connection can define the pressure at this connection but not the flow. A pressure input connection is RED. A pressure output connection is GREEN.

It makes sense to connect these connectors in an alternating fashion so that one defines the flow and other defines the pressure and they do not overwrite each other. Indeed, if a pressure connection is connected to a flow connection and vice versa, the simulation will give an alarm and point out the blocks that are causing the issue.

The pressure/flow objects have either all pressure connections or all flow connections but some of the boundary objects can have mixtures of them. Since the pipes and valves can calculate their flows as a function of the pressures at either end, they have flow connections. Conversely, the pressure nodes calculate their pressure as a function of the flows going into them.

Another way of stating this is that all flow objects (those with flow connections) must either be connected to a node or a boundary object that is a pressure source such as a tank inlet or outlet or a source set to pressure source mode.

Table 1: Pressure Flow Objects

Object	Type
Pipe	Flow
Valve	Flow
Pump	Flow
Compressor	Flow
Aftercooler	Flow
Node	Pressure

Table 2: Boundary Objects

Object	Type
Tank	Pressure (Except a tank overflow, which is a flow connector)
Sink	Can be a flow object or pressure object, which can be selected from Dialog box
Source	Can be a flow object or pressure object, which can be selected from Dialog box

Appendix B. Scan Classes

IDEAS allows the user to specify different scanning rates for different classes of objects. The Discrete/Continuous Executive object has a dialog box that looks like this:

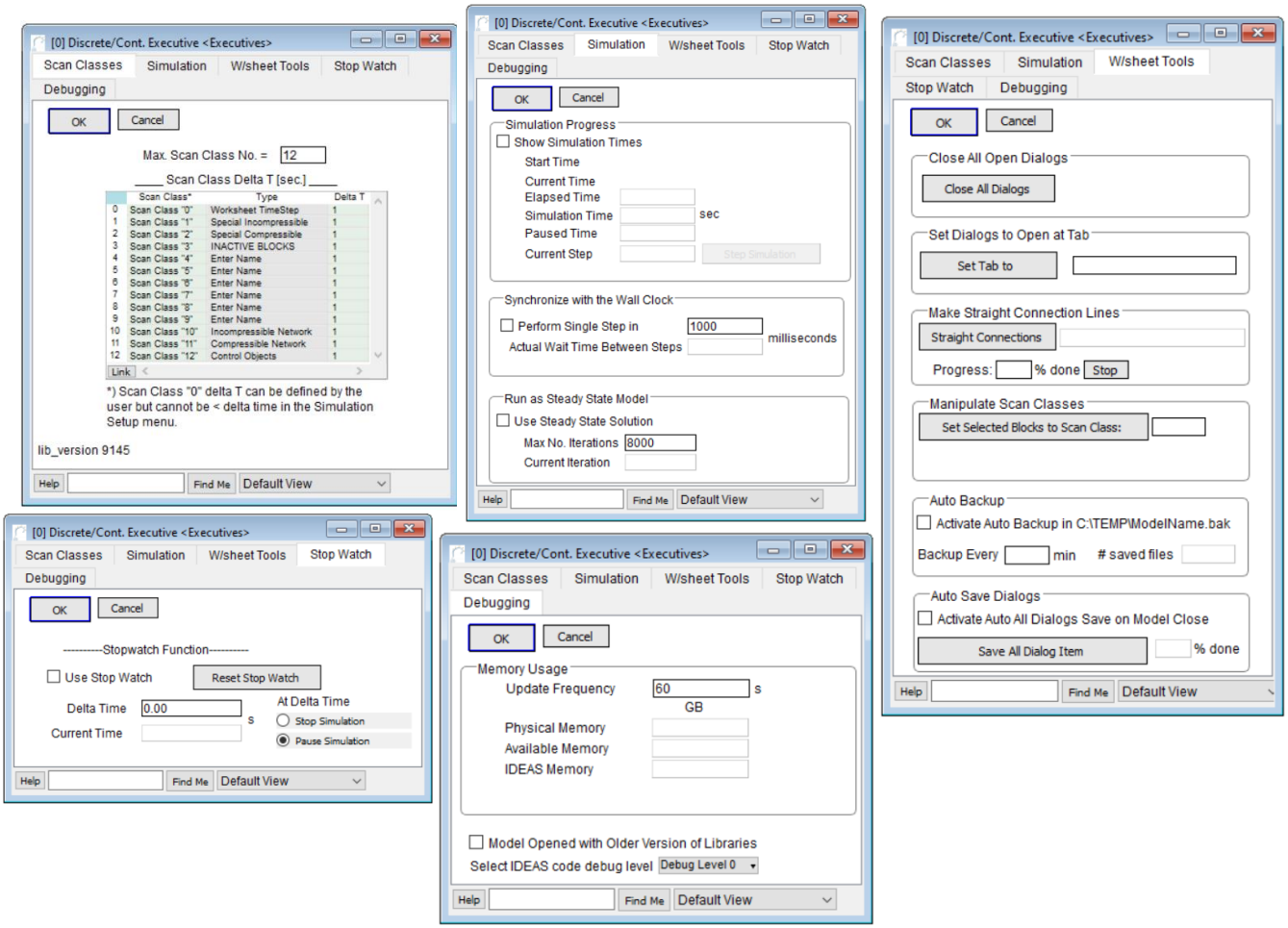


Figure 64: Discrete/Continuous Executive Dialog Box

There are two types of scan classes: Selectable and Dedicated.

B.1 Selectable Scan Class

Boundary objects, i.e., those that are not included in the global solver solution (see Appendix A for further

explanation) can be assigned to one of thirteen separate scan classes. Although they are all defaulted to be equal to one (1) second, each of the selectable scan classes can be set to a different number. Scan class 0 delta T can be defined by the user but cannot be less than delta time in the Simulation Setup menu. Each boundary object should default to scan class zero so that it runs under the overall delta time provided by the IDEAS executive. This scan class can be changed to scan class 1 through 12, allowing the object to run at a different rate from the other objects on the worksheet.

B.2 Dedicated Scan Class

The dedicated scan classes are scan classes reserved for special object types. Below is a table of the dedicated scan classes.

Table 3: Scan Class Types

Scan Class Name	Scan Class Type
Scan Class “1”	Special Incompressible
Scan Class “2”	Special Compressible
Scan Class “10”	Incompressible Network
Scan Class “11”	Compressible Network
Scan Class “12”	Control Objects

For example, this means that every regular, incompressible network object (e.g. Pipe w/ Valve) will automatically define the object with a scan class of 10. Dedicated scan classes help ensure that all objects within the same scan class type execute at the same frequency.

Objects that perform time related functions like integration can be adversely affected by a different scan class other than the pressure/flow objects’ scan class.

Example: Within a tank object, the scan class is changed from scan class 0 to scan class 1 where scan class 1 is set to 4.0 in the Discrete/Continuous Executive block.

In this example, when the tank runs, it will integrate the flows only every 4 seconds. The varying flow values during that 4-second time step will not be used within the integration. If the flow does not change within that 4-second time step, there is no error, but if the flow changes the error can be significant:

Tank Integration Example:

Scan Class 0 = 1.0 second

Scan Class 1 = 4.0 second

Incompressible

Scan Class = 1.0 second (flow update time)

Table 4: Scan Class Example

Time (s)	Flow into Tank (kg/s)	Tank Contents (Scan Class 0)	Tank Contents (Scan Class 1)
1	10	10.0 (10 x 1 s)	40.0 (10 x 4 s)
2	15	25	40
3	20	45	40
4	25	70	40
5	30	100	160
6	35	135	160
7	40	180	160

Appendix C. Creating Hierarchical Objects, New Libraries, Blocks, and Objects

This appendix will go into details of how to create H-blocks, create and compile new libraries, and build new objects.

The user may remember that a hierarchical object (H-block) is a special object that contains other objects (or even other hierarchical objects) connected together like a model.

A hierarchical object is unique: it has some characteristics of an object and some characteristics of a model worksheet.

Unlike other objects, hierarchical objects are saved directly in the model as copies. This characteristic allows them to be treated much like a copy of a portion of the model. The user can copy a hierarchical object to another part of the user's model and make changes to its hierarchical window without affecting the original hierarchical object. This is also known as physical hierarchy.

- If you modify a hierarchical block's submodel in its worksheet (accessed by double-clicking the icon), those changes only apply to that block. Changing a hierarchical block's submodel is similar to changing parameters in a regular block's dialog: the changes affect only that instance of the block on the worksheet and are saved with the model. This is true even for hierarchical blocks that were originally saved in libraries.
- If you modify the structure of a hierarchical block that is not saved in a library, those changes also only apply to that block. This has the same result as modifying a hierarchical block's submodel in the worksheet window. For example, you can make several copies of that hierarchical block in a model, but when you change one of the copies, the other blocks remain unchanged.
- If you modify the structure of a hierarchical block that has been saved in a library, you can choose how those changes should be reflected:
    - Only to this instance of the block on the model worksheet (choose Save Changes to This Block).
    - Also in the master block in the library, which only affects blocks placed in the model from the library after the change has been made (choose Also Save to Library).
    - Also in all instances of the block in open models (note that this does not affect models that are not open at the time); this is also called pure hierarchy (choose Also Update Blocks in Open Models). This option will also update the object in the library.

Upon opening a hierarchical object structure, the user sees a window where there are tabs for Icon, Worksheet, and Help. The worksheet tab contains another view of the layout pane where the user builds a new hierarchical object or makes changes to the existing one. Hierarchical structure windows are discussed further in "Building a New Hierarchical Object" within section C.2 - Making a Selection a Hierarchical Object.

When the user opens a hierarchical object by double-clicking on it, instead of seeing a dialog, a layout of the submodel in the hierarchical window is seen. For example, an open hierarchical object would look like this:

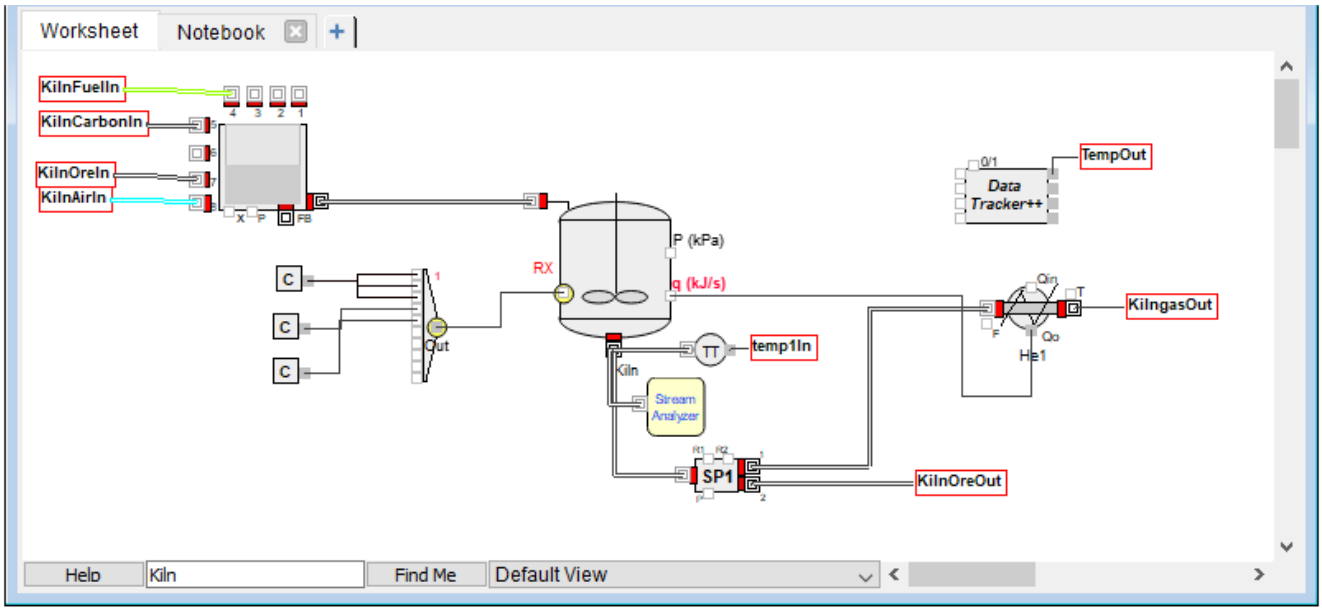


Figure 65: Hierarchical Object Layout Pane

C.1 Two Methods for Creating Hierarchical Objects

As discussed below, there are two methods for creating hierarchical objects:

- Select some objects from a model and choose Make Selection Hierarchical from the Model menu (bottom up approach).
- Create a new hierarchical object with the New Hierarchical Block command from the Model menu (top down modeling).

If the user wants to use hierarchical objects to reduce the complexity of an existing model, probably the first method will be used more often. On the other hand, if the user is building a model from the ground up, the user may want to create new hierarchical objects from scratch as well. Both methods are described in detail below.

No matter which method the user uses to create them, all hierarchical objects have the following in common:

- The user can copy them to other areas of the model or to other models and use them like any other object.
- The user can change the settings of an object within a hierarchical object by double-clicking on the hierarchical object, then double-clicking on the desired object's icon.

The user can treat hierarchical layout windows like model windows. The user can add objects from a library, create hierarchical objects, draw items, type labels and other text, clone dialog items onto them, and so on.

The user can modify a hierarchical object's icon, connectors, or help text by holding down the Alt key and double-clicking on the object's icon. This opens the hierarchical object's structure window.

C.2 Making a Selection a Hierarchical Object

To make several objects into a single hierarchical object:

Select the objects and any desired draw items by dragging over them or by holding down the Shift key and clicking on each of them.

It should be noted that the selection tool the user uses determines what is selected. See note below for information about including named connections in the selection.

Choose Make Selection Hierarchical from the Model menu (Ctrl + M). The user can also select the blocks, right-click, and then select “Make Selection Hierarchical. The user will see the following dialog:

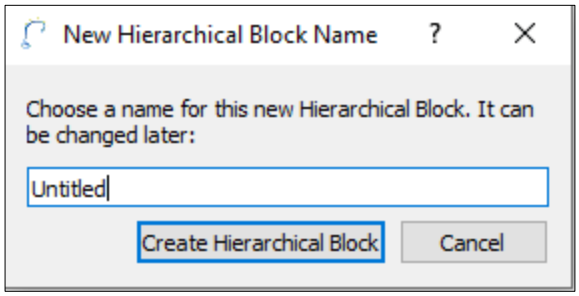


Figure 66: Make Selection Hierarchical Dialog

Enter a descriptive name for the object and click the Create Hierarchical Block button.

Note: This action cannot be undone. If the user accidentally includes more objects than the user intended in the hierarchical object, the user must remove them from the hierarchical object and put them back in its parent worksheet using the Cut and Paste commands from the Edit menu.

When the user uses the Make Selection Hierarchical command, IDEAS makes all the connections for the user and replaces those objects in the model with the new hierarchical object, including a default icon. To verify that this is so, double-click on the hierarchical object’s icon and look at the submodel. The window that holds the submodel is shown below:

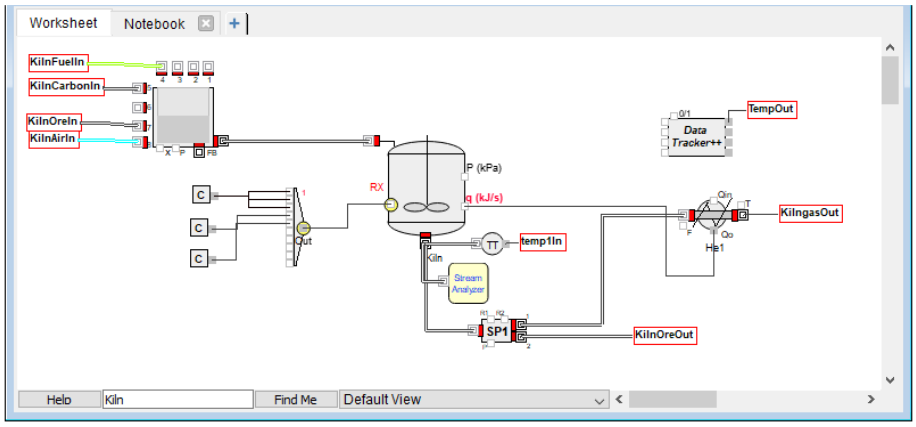


Figure 67: Hierarchical Object Layout Pane

Note: Named connections only work on one level in a hierarchical model. The data will not be transferred between levels. This means that a named connection on one level will not communicate with a corresponding named connection in an object at a lower or higher level. If the user includes named connections inside a hierarchical object, the user will have to provide a connector if the data is supposed to go outside the hierarchical object.

The user can alter aspects of the object such as moving the connectors, choosing a different picture, or creating a new design using the drawing tools. See section C.9 1.1.1.1C.9 for more details.

Note: When the user selects multiple objects and chooses the option to make a selection hierarchical, the hierarchical object is saved only in the model. If the user wants to save it in a library, the user must use the Save Hierarchical Block to Library As command.

Building a New Hierarchical Object

To build a new hierarchical object, the user must have a model window open. When the user selects the New Hierarchical Block command in the Model menu (Ctrl + U), IDEAS prompts the user for a name, and then opens up a structure window for the user.

The structure window for a new hierarchical object is shown in Figure 68 below:

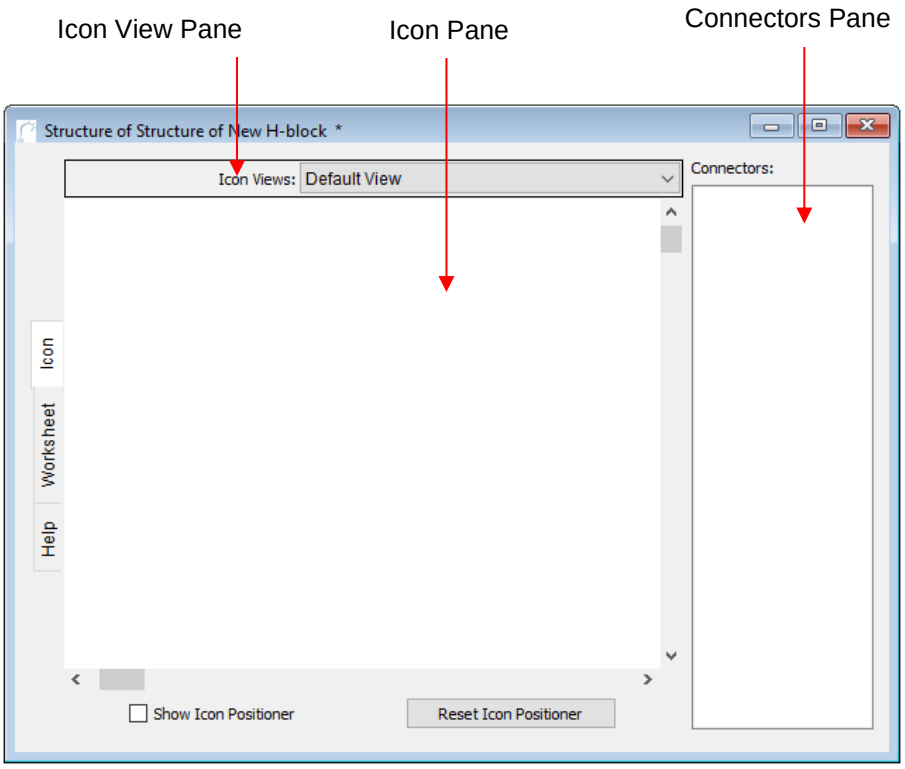


Figure 68: Structure Window for Hierarchical Objects

The hierarchical block's structure window has three tabs on the left side (Icon, Worksheet, and Help).

The Worksheet tab is where the submodel for this hierarchical block will be built; this is what appears when you double-click a hierarchical block's icon.

The Help tab is for entering any comments or descriptions about how the block works.

The Icon tab, shown above, is divided into three work areas or panes for creating the block's icon and connectors:

- The area at the top left of the Icon tab is the Icon View pane, for creating different views for the icon, such as left and right view.
- The Icon pane in the middle is for designing the block's icon and adding connectors.
- The Connectors pane on the right displays the names of the input and output connectors for the block. It also appears also on the Worksheet tab and can be used to connect to the blocks. We can rename the connectors either in the worksheet tab or in the connectors pane in the icon tab.

The user can save it to a library, if desired. Connect the new hierarchical object to the other objects in the model

Note: The Icon Positioner is only used if the position of the icon needs to be adjusted.

C.3 Building the Submodel

To build a submodel in a hierarchical object, use the same methods used when the user built a model in Section 3.18 Building a Model. Use the Library menu to add objects to the layout pane, or click on the object in the library window and then click on the worksheet to place the object on the worksheet. Connect objects in the normal fashion.

If the user prefers, shortcuts (Copy and Paste) from the Edit menu, can be used to copy a portion of an existing model into the layout pane to be used as a submodel. Once the submodel is built, the user can modify the layout or enhance it with drawing objects or text as discussed in Section C.9 Modifying Hierarchical Objects.

C.4 Modifying the Icon and Setting up Connectors

The hierarchical object starts with a default icon. The user can modify this icon, for example, by changing its shape or color, or the user can delete it and create a new one. To do this, draw an icon with the drawing tools in the toolbar, or paste a picture in from another program.

Note: If the Icon pane is not visible, the user needs to go to Tools and select Icon (check mark). The icon pane will now appear on the right hand side of the window.

Since the hierarchical object needs to be connected to the objects in the model, the user must add the appropriate connectors. Some hierarchical objects have both input and output connectors, while others have just one kind. Also, hierarchical objects can have either value or item connectors, or both, or no connectors.

When the hierarchical structure window is the working window, the toolbar has additional tools from what the user saw in Section 4 Special Features.

Figure 69: Structure Window Toolbar

The  connector symbol on the palette is used to add animation objects. The next four connector types are scalar (), stream array(), universal (), and user-defined ().

There are four steps to setting up a connector to a hierarchical object: decide the type of connector to add; add the connector; determine if the connector should be an input or an output connector; and connect the objects in the layout pane to the connector.

First, decide which type of connector to use. The user will typically use the scalar or stream array connector. Please note that the connector type must match the main input and output connectors for the submodel in the

layout pane.

To add a connector, click on the connector in the toolbar; then click in the icon pane at the desired position near the edge of the hierarchical object's icon:

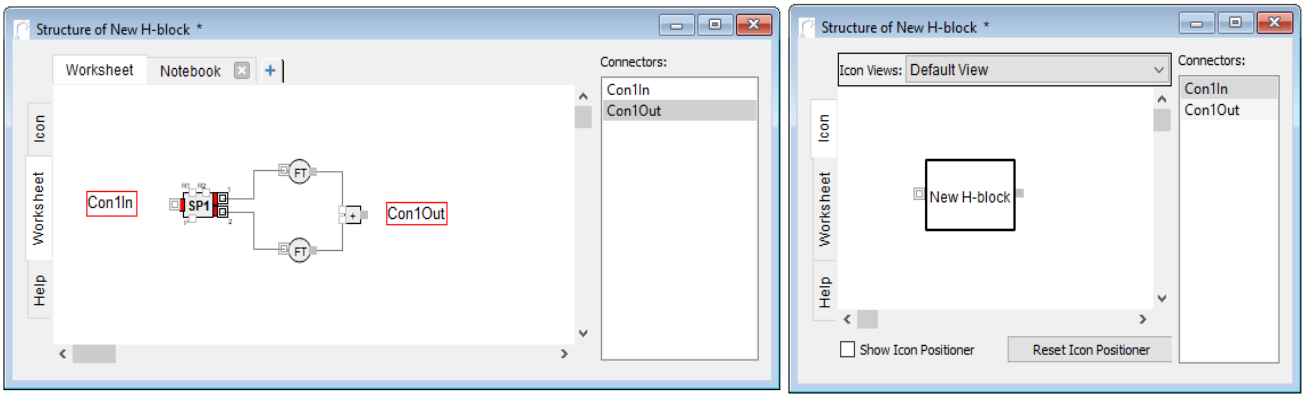


Figure 70: Output Connector Added to New H-block Icon

This creates the connector in the icon pane, lists it in the connector pane, and adds a connector text object (very much like a named connection) in the layout pane.

Note: If the user chooses the wrong type of connector, just click on that connector on the icon pane to select it. Now click on the correct connector type in the toolbar. The connector will change to the new type.

Every connector has a unique name. The ending of the name of the connector ("In" or "Out") defines whether it is an input or output connector. When the user adds connectors to the icon, they are all initially input connectors. To make one of these connectors an output connector, change its name to something that ends with "Out." The user can change the connector name as desired, but the name must end in "In" or "Out." To change the name of a connector, select the connector name in the connector pane. Type a new name or edit the name, and press the Enter key or click anywhere else in the connector pane to save the edited name.

Once the user has added the connector to the icon, the user needs to connect the connector text object in the layout pane to the appropriate objects. The user does this the same way as connecting a named connection to an object in the model window: draw a line from the connector of an object in the layout pane to the connector text. When the line thickens, click on the text object to make the connection. Named connections are described fully in Section 3.18 Building a Model.

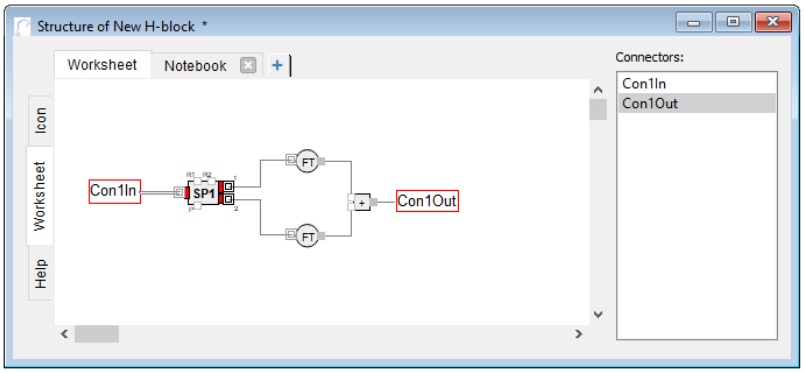


Figure 71: Hierarchical Connector Connected to Submodel

C.5 Saving the Object

The user can save the new hierarchical object so that it is saved just in the model or in a library. If the user closes the structure window by clicking on its close box (or by choosing Close in the File menu), a pop-up dialog box will appear and allow the user to choose how they would like to save the object. See section C.7 - Saving Hierarchical Objects below for more details.

C.6 Connecting the Hierarchical Object in the Model

To connect the finished object to the model, first close the structure window if it is open. Then, simply connect the connectors on the hierarchical object to other connectors in the model, just as the user would with any other object.

C.7 Saving Hierarchical Objects

Once the user has built a hierarchical object, the user can save it just in the object, in all the objects in the open worksheets, or within the library.

The user can make hierarchical objects that are saved with the model but which do not exist in libraries. When the user makes a selection hierarchical, the hierarchical object is automatically saved with the model when the model is saved. If the user makes a new hierarchical object, or makes changes to the layout pane window of an existing hierarchical object, then click its close box to save it, it exists only in the model. The user can copy such a hierarchical object to other parts of the model or even to other models using the Clipboard. Each instance of the object in the model can then be made unique by modifying it as described in the section below. When the user changes one hierarchical object, the other objects do not change at the same time.

The user can save a hierarchical object in a library, but what is saved in the library is only a "snapshot" of the hierarchical object at the time the user saved it. If the user later modifies that hierarchical object on the model worksheet, the changes may or may not be reflected in the master object in the library. For example, if the user

modifies the submodel of a hierarchical object from a library, the changes will not be saved in the master library object unless the user tell IDEAS to save those changes, as described below.

To save a hierarchical object in a library or to save changes made to a hierarchical object to its master object in a library, follow the method mentioned below:

If it is not already open, open the structure window of a hierarchical object by holding down the Alt (Option) key as the user double-clicks the object's icon in a model window, or right-click the object and choose Open Hierarchical Block Structure. (The structure window is the active window when the user is building a new hierarchical object or when the user is modifying a hierarchical object's icon or connectors.)

Choose *Save Hierarchical Block to Library As…* from the File menu. In that dialog, choose a library for the hierarchical object and install the hierarchical object in the library. Then close the hierarchical object and choose one of the save options, as discussed in C.9 "Modifying Hierarchical Objects."

It is strongly recommended that the user does not save hierarchical objects in the libraries that come with IDEAS; instead make a new library for the user's hierarchical objects.

A hierarchical object that is saved in a library has its name listed in the library with a greater than (>) symbol.

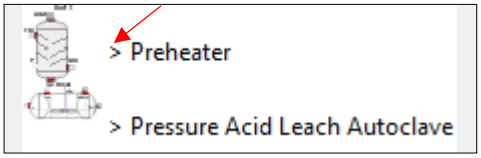


Figure 72: Indication for H-block

Like other objects, hierarchical objects that are saved in libraries list the names of the libraries in their structure window. If no library is listed, the hierarchical object is not saved in a library. The user can also tell if a hierarchical object is saved in a library by selecting the object in the model window and selecting Properties from the File menu.

If the user modifies a hierarchical object that is saved in a library, the changes may not be saved to the master object in the library. Saving changes to a hierarchical object that is in a library is discussed in section C.15 Results of Modifying Hierarchical Objects.

C.8 Ways to Modify Hierarchical Objects

How the user modifies a hierarchical object depends on what it is that the user wants to modify. Typically, the user might want to modify the settings of the objects in the submodel, the layout and appearance of the submodel, or the hierarchical object's icon, connectors, or help:

The user can change settings for a sub-model's object by double-clicking on the object's icon in the layout pane window. The user can also clone dialog items onto the layout pane as discussed in Section 3. The user usually

accesses the layout pane window by double-clicking on the hierarchical object's icon, although the user can also access it in the hierarchical structure window.

The user can modify the submodel as the user would any model: add additional objects, create hierarchical objects, type text and add drawing objects, or move objects and connectors within the hierarchical layout pane window. The user does this by double-clicking on the hierarchical object's icon to access its layout window. Then, use the Library menu and toolbar to add objects, text, and so on. This is the recommended method to use if the user wants to modify the submodel only in this one instance of the hierarchical object.

The user can modify the hierarchical object's icon, connectors, or help in the object's structure window. If it is not already open, open the structure of a hierarchical object by holding down the Alt key as the user double-clicks the object's icon in a model window, or select the object and choose Open Hierarchical Block Structure in the Model menu.

The user can also use this method to modify the submodel in the layout pane, although this is more common when the user wants to save the object in a library. The procedures for making changes are the same as for building a new hierarchical object.

Note: If the hierarchical object is in a library, the user cannot modify its structure by double-clicking on its icon in the library window. The user must work directly on the worksheet, and the user must use a hierarchical object originally copied or placed from the library menu or library window.

C.9 Modifying Hierarchical Objects

The user can add many enhancements that will improve the hierarchical objects:

- Add text, drawing objects, and pictures (layout window or structure window must be open).
- Clone dialog items (layout window or structure window must be open).
- Change the icon (structure window must be open).
- Rename the object (structure window must be open).
- Add help text (structure window must be open).

To access the layout pane directly, double-click the hierarchical object in the model. To access the hierarchical structure window, hold down the Alt (Option) key while double-clicking the hierarchical object in a model window, or select the object and choose Open Hierarchical Block Structure in the Model menu. The user can also access the layout pane by opening the structure window, although this is less common.

C.10 Adding Text, Drawing Objects, or Pictures to the Layout Pane

The user can add text, drawing objects, and pictures to the hierarchical object just as the user does in a model. To do this, open the layout window by double clicking on the hierarchical object's icon. Then use the tools in the toolbar to enhance the submodel.

C.11 Cloning Dialog Items to the Layout Pane

The user can clone dialog items from the dialogs in submodel objects directly to the layout pane. This lets the user see the items when the user double-clicks the hierarchical object's icon. The method for doing this is identical to the normal cloning method the user learned in Section 3.

C.12 Changing the Icon

The user will probably want to change the icon for the hierarchical object, since IDEAS starts with just a plain rectangle. The user has the option to change and edit the icon to better suit the purpose of the hierarchical object.

Open the structure window by holding down the Alt (Option) key double-clicking on the object's icon or select the hierarchical object and choose Open Hierarchical Block Structure in the Model menu.

Click on the icon in the icon pane and change or delete it (but be careful not to delete the connectors).

Use any of the drawing tools to make the icon. The user can also paste pictures from outside of IDEAS into the icon pane.

Note: It is very important that the user does not delete any connectors. If the user does so, IDEAS will warn the user. If the user accidentally deletes a connector, the user can undo it or the user can add another one, but the user must then check the object's connections in the model.

The user can change the color of the default icon without deleting it; just select it and choose new colors. If the user wants a different icon shape, the user must delete the default icon and draw a new one. See Section 3 for information on how to use the drawing tools. Drawing in the icon window is just the same as drawing in the model window.

C.13 Renaming the Object

When the structure window of a hierarchical object is open, the user can rename the object by choosing Rename Block from the Develop tab of the IDEAS application.

C.14 Adding Help Text

The user can change the help text by editing in the help tab in the structure window. The user can type in text or use Copy and Paste to move it from other objects. The user can change the style of text in the help pane by selecting it and giving commands from the Text menu.

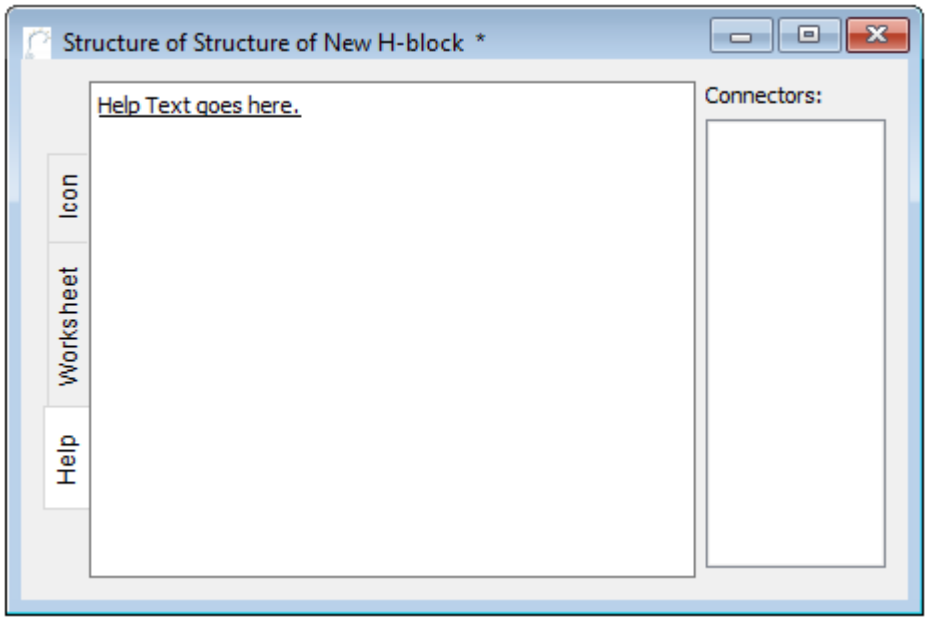


Figure 73: H-block Help Tab

C.15 Results of Modifying Hierarchical Objects

There are different results when the user modifies hierarchical objects, depending on whether the user modifies its layout pane window or structure window, and whether the object is saved in a library:

If the user modifies a hierarchical object's submodel in the layout pane window, those changes only apply to that object. Changing a hierarchical object's submodel is similar to changing parameters in a regular object's dialog: the changes affect only that instance of the object on the worksheet and are saved with the model. This is true even for hierarchical objects that were originally saved in libraries.

If the user modifies the structure of a hierarchical object that is not saved in a library, those changes only apply to that object. This has the same result as when the user modifies a hierarchical object's layout pane window.

For example, the user can make several copies of that hierarchical object in a model, but when the user changes one of the copies, the other objects remain unchanged.

If the user modifies the structure of a hierarchical object that is in a library, the user can choose how the user wants those changes reflected:

- Only in this instance of the object on the worksheet (Save Changes to This Block).
- Also in the master object in the library (only affects objects placed in the model from the library after the change has been made) (Also Save to Library).
- Also in all instances of the object in open models (note that this does not affect models that are not open at

the time); this is also called pure hierarchy (Also Update Blocks in Open Models).

For example, when the user closes the structure window of a hierarchical object from a library, IDEAS prompts the user with the following dialog:

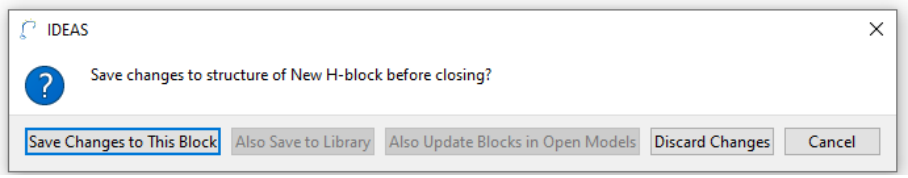


Figure 74: Save Changes Dialog Box

C.16 Closing the Structure Window

Choose "Save Changes to This Block" to save the changes just in this object or "Also Save to Library" to make the changes appear in the library holding the master object. "Also Update Blocks in Open Models" makes the changes to all copies of that object that came from the library and are used in any open models but not to models that are closed.

C.17 Creating New Libraries and New Blocks / Objects

The user can create new libraries and new blocks in IDEAS using the ModL Programming language. For detailed information on Creating Blocks, please refer to the *ExtendSim User Manual* and the *ExtendSim Developer Reference*.

Appendix D. Sensitivity Analysis

As discussed in Section 3.23, Running the Model, sensitivity analysis allows the user to conduct controlled experiments to explore how much of an impact a particular parameter has on model results. IDEAS's sensitivity analysis features make it easy and convenient to specify the parameter the user wants to investigate and settings to use for the analysis.

D.1 Introduction to Sensitivity Analysis

Sensitivity analysis works with all numeric parameter entry items (the rectangular fields in the objects' dialogs in which the user keys in numbers). It will also work with clones of those numeric items. The user enters the sensitivity settings for a particular dialog parameter using a special key (Refer to Section D.2) as the user clicks on the parameter or its clone. The user can add sensitivity to as many dialog values as the user needs. However, it is recommended that the user only vary one or two dialog values at a time to avoid confusion with the analysis.

Once the user has "sensitized" a parameter, the user specifies how many simulation runs the user wants. When the user runs the simulation multiple times with the Use Sensitivity Analysis command checked, the user sees the results of varying the parameter value over the settings the user has chosen.

The Open Sensitized Blocks command from the Model menu shows the user all the dialogs for objects that have sensitivity settings. This is convenient if the user has entered sensitivity settings for many parameters in a large model.

D.2 Steps for Using Sensitivity Analysis

Open the dialog of the object that has the value the user wants to vary.

Hold down the Control key (CTRL-Windows) while clicking once on the desired numeric entry, or click on the desired numeric entry and choose the Sensitize Parameter command in the Model menu.

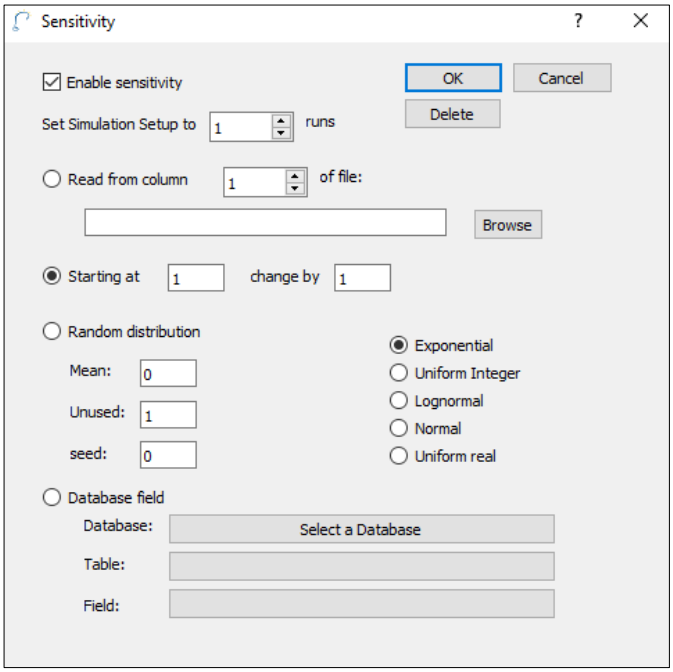


Figure 75: Sensitivity Parameter Dialog Box

Be sure the "Enable Sensitivity" box is checked. As discussed below, this checkbox allows the user to temporarily turn sensitivity analysis off for this parameter without losing the settings.

Specify the number of simulation runs needed for the analysis. The user can do this either in the Sensitivity Setup dialog or in the Simulation Setup dialog. Each setting controls the other, so that the last value selected in either of them controls the number of runs.

Choose the method of sensitizing the parameter (a file, a specified range, or a random distribution, as described below) and enter any needed values. The choices are described in Section D.3 Specifying the Sensitivity Method and number of Runs.

Be sure the Use Sensitivity Analysis command from the Run menu is checked. If not, choose the command so that a check mark appears next to it.

Run the simulation.

When using sensitivity analysis, the user can investigate the effect of changing one parameter. The user usually plots the resulting values for that area of interest on one of IDEAS's plotters. Although the user can use any of the standard plotters (such as the Plotter I/O or the Plotter Discrete Event), it is more common for the user to use a MultiSim plotter (to show up to four runs at a time in one plot window).

D.3 Specifying the Sensitivity Method and Number of Runs

In the Sensitivity Setup dialog, the user can specify if a sensitized parameter will change incrementally, randomly, or from a list in a file. The four choices are:

1. Option: Read from column… of file

Assigns the values from a text file. This is the option the user will most likely use when performing ad hoc experiments. Use this if the user has the desired values in a text file that was created in IDEAS or in another program. If the file has more than one column separated by tab characters, the user specifies the desired column. Click on the Browse button to enter a filename to be read from.

2. Option: Starting at ... change by...

Specifies the starting value and the amount of change. By default, the starting value is the same as the parameter's value in the dialog. The user can increase the variable with a positive number, or decrease it with a negative number.

3. Option: Random Distribution

Uses a random distribution to set the parameter. This is an easy way to make a single value in the model change randomly over many simulations while keeping the value constant within a single run. Choose from one of the five types of distribution and enter the distribution parameters in the options to the right of the distribution. The seed is the number to use for the random number generator. As in the Simulation Setup dialog, BLANK or 0 for the seed is random.

The user can specify the number of simulation runs either in a Sensitivity Setup dialog or in the Simulation Setup command's dialog.

Note: The last number of runs entered in either dialog determines how many times the simulation will run.

4. Option: Database Field

Assigns values from the fields in an ExtendSim database. Starting with the first record of the specified field for the first run, this option uses the value of each successive record for subsequent runs.

D.4 Turning Sensitivity On and Off

The user can control sensitivity globally (for the model as a whole) or locally (at the dialog parameter level): Use a menu command to turn sensitivity analysis on and off for the model as a whole. Enable, disable, and delete sensitivity settings for a particular parameter in the dialog in which the parameter appears. When the user enters

sensitivity settings for a value, sensitivity analysis is enabled as long as the "Enable sensitivity" box is checked in the Sensitivity Setup dialog. If the user unchecks the box, the user temporarily disables a dialog value's sensitivity so that the user does not have to reenter the number for subsequent analysis.

If sensitivity analysis is active for the parameter (that is, if the "Enable sensitivity" choice is checked, as discussed above), the frame is green or solid black. If the sensitivity analysis is inactive for the parameter or if it is turned off for the model as a whole, the frame is red or dotted grey. The Open Sensitized Blocks command (in the Model menu) shows the dialogs of all objects with sensitized parameters, regardless of whether or not the "Enable sensitivity" box is checked.

To remove sensitivity settings from a parameter (as compared to temporarily disabling the settings by turning off the parameter's "Enable sensitivity" box), open the Sensitivity Setup dialog (or select the Sensitize Parameter command from the Edit menu). Then click the Delete button.

Note: Editing a sensitized parameter in an object's dialog disables the sensitivity settings for that object. When this happens, IDEAS automatically unchecks the "Enable sensitivity" choice. IDEAS assumes that if the user is editing the value, the user wants to use that new value, not the one that was entered in the Sensitivity Setup dialog. If the user wants to turn off sensitivity analysis for a number for the foreseeable future, open that item's Sensitivity Setup dialog and click the Delete button. This will help prevent accidentally changing the value in a future run of the simulation.

D.5 Reporting the Results

In addition to Multisim-Scan Plotter, IDEAS's reporting and tracing features are useful when analyzing output after using sensitivity analysis.

The Report feature shows final values and the Trace feature shows values at each step or event.

D.6 Multi-Dimensional Scenarios

The user can enable sensitivity on more than one item at a time. For instance, the user may want to vary the values of two Constant objects and see the interaction between the two items. If the user sets the sensitivity for the parameters with the "Starting at" option, both values will increment at the same rate. For instance, if the user has one parameter that starts at 5 and increments by 1, and the second parameter starts at 100 and increments by 50, and the user runs the simulation seven times.

The value pairings will be:

Table 5: Multi-dimensional Scenarios Values Pairs

Run No.	Variable 1	Variable 2
0	5	100
1	6	150

2	7	200
3	8	250
4	9	300
5	10	350
6	11	400

Often, however, the user wants to look at all the possible combination of the two (or more) variables. In this example, the user would want to run the model 36 times, with the following combination:

Table 6: Multi-dimensional Scenarios Values Pairs

Run No.	Variable 1	Variable 2
0	5	100
1	5	150
2	5	200
3	5	250
...	...	...
7	6	100
8	6	150
...	...	...
35	11	400

In order to perform this kind of multidimensional analysis, the user needs to get the values from a file. The most convenient way to do this is to create a file that has two columns separated by a tab character with all the desired pairings. For instance, the file for this example would start off: 5 100, 5 150, 5 200, ...

In the Sensitivity Setup dialogs, the user chooses "Read from file" and chooses the file name. For the first variable, enter "1" for the column number; for the second variable, enter "2."

Appendix E. Creating New Components

E.1 Using the MP Assembler Object to Create New Components

Creating a new component, as a component record file, requires the use of the MP Assembler object from the Material Properties library (MP Tools submenu).

1. Place an MP Assembler object on an IDEAS worksheet from the Material Properties library.

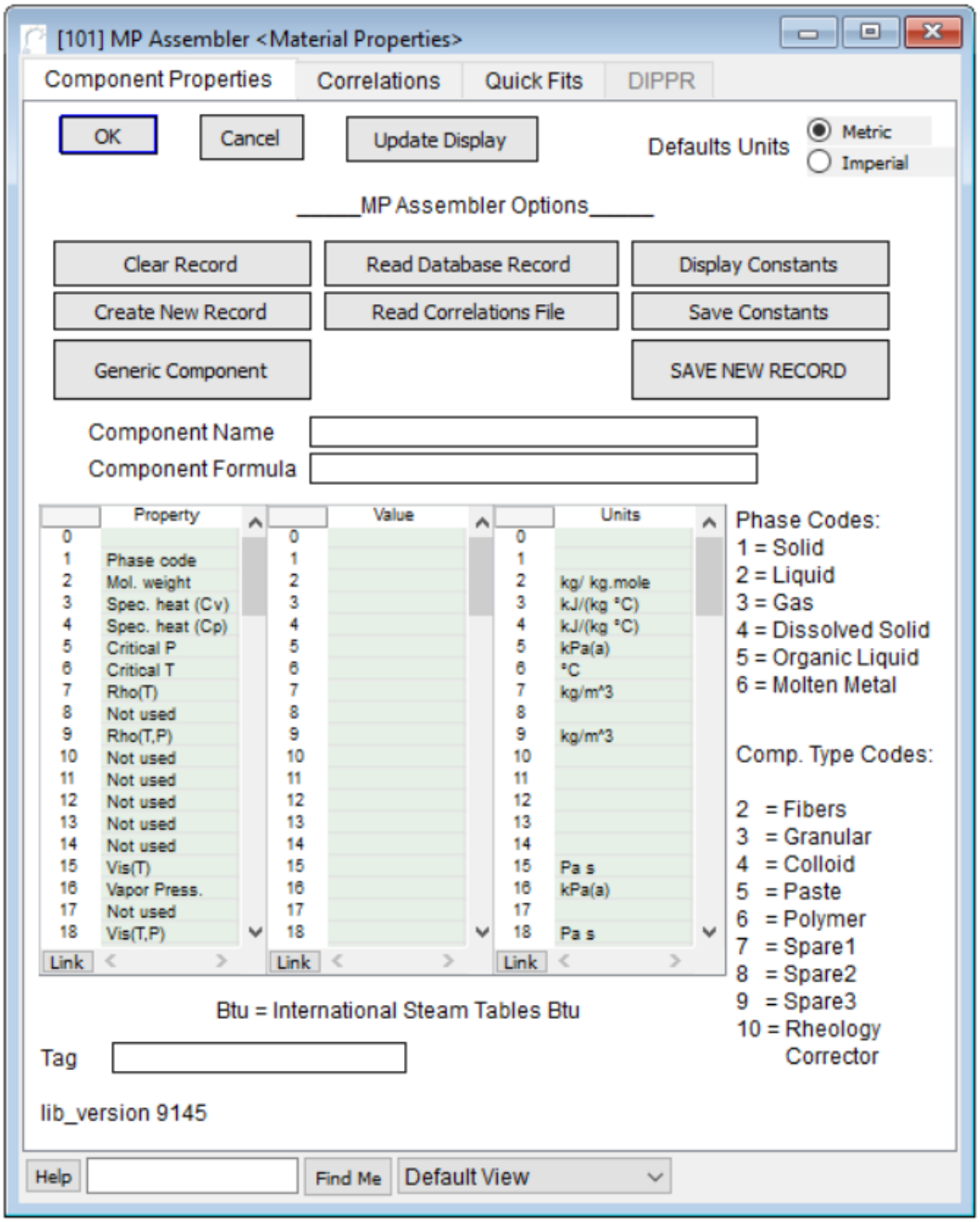


Figure 76: MP Assembler: Component Properties Tab

1. In the Component Properties tab, select units (Metric or American). Enter the name of the new component in the field Component Name, for example, NewComp_s. Use the appropriate suffix to indicate the phase of the new component: _s for solid, _l for liquid, _g for gas, _aq for dissolved solid, _or for organic, _mo for molten metal.

2. Enter the chemical formula, if known, in the field "Component Formula," e.g., FeCl2.
3. Press "Create New Record" button.
4. The tables Property, Value, and Units of the Component Properties tab are used to enter various property values. Enter values into the Value table in the units of the Units table.
    - Regarding Constant Values and Correlations: During simulation, IDEAS objects calculate various properties of their materials. If a correlation has been entered for a property, that correlation will be used – see also “Adding Correlations”. If no correlation is available, the value entered in the Value table will be used.

Regarding Required Data: During simulation, IDEAS will use component data as required by specific calculations. If the required data is not provided, IDEAS will use the value 0, which will, in many cases, cause errant, or no value, results. Not all data shown in the Property table are necessarily required for simulation. For example, viscosity data is not needed for a steady-state model because only dynamic models use viscosity data to calculate flow through pipes and ducts. If limited component data is available, the user may want to first load generic (water) data into the Values table. If desired, simply press the Generic Component button to load this generic liquid data into the Value table.

Then continue by entering known data, replacing any generic data in the Value table. Typical required data is listed below:

- Property 1 "Phase Code" - Component Phase
- Property 2 "Mol. Weight" - Molecular Weight
- Property 3 "Spec. Heat (Cv)" - Specific Heat Capacity (not used for non-gases)
- Property 4 "Spec. Heat (Cp)" - Specific Heat Capacity
- Property 7 "Rho(T)" – Density
- Property 9 "Rho(T,P)" – Density (not used for non-gases) Property 15 "Vis(T) – Viscosity
    - Water viscosity = 0.000993 Pa s (0.993 cP)
    - Air viscosity = 1.867e-05 Pa s (0.01867 cP)
- Property 18 "Vis(T, P)" - Viscosity (not used for non-gases)
- Property 27 "Enth formation" - Enthalpy of Formation at 25 degrees Celsius

Press the "Save Constants" button to save the data temporarily. This does not save the component record. Proceed to "Saving Component Records" at the end of this document if no correlations are being added for this component.

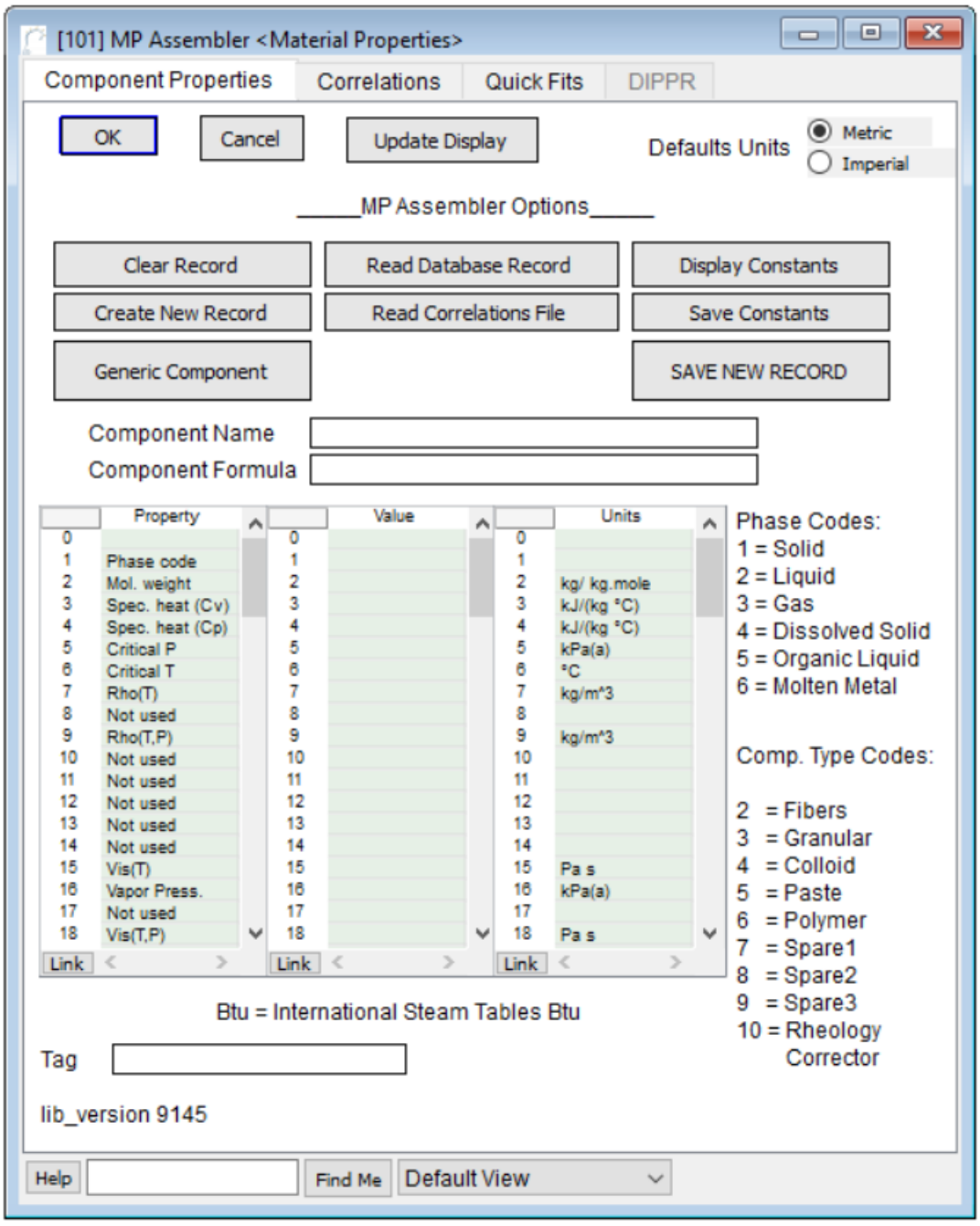


Figure 77: MP Assembler: Component Properties – Ready to Save Record

E.2 Adding Correlations

Correlations may be added to a component record for various component properties. Some property correlation

parameters may be added directly in the Quick Fits tab of the MP Assembler object. These correlations are described here. Other correlations may be created from raw data using the IDEAS "MP 1 Variable Curve Fitter" and "MP 2 Variable Curve Fitter" objects.

E.3 Enthalpy Correlations

If an enthalpy correlation is not entered for a component, IDEAS will use the specific heat and enthalpy of formation values to calculate the enthalpy of that component at different temperatures. For components with constant specific heat capacity, this method is exact. However, most components have a non-constant specific heat capacity. For better accuracy, and if enthalpy correlations are available, enthalpy correlations may be entered into the component record file as described. Do not enter specific heat correlations – they are not used by IDEAS – enthalpy is used instead.

The IDEAS MP Assembler object uses the following correlation for (specific) enthalpy as a function of temperature:

Enthalpy = A + B*T + 0.001*C*T^2 + 100000*D/T + Hf

Where:

Enthalpy = Specific Enthalpy in kJ/kg, kJ/kmole, or kcal/kmole
T = Temperature in degrees Kelvin
Hf = Enthalpy of Formation at 25 degrees Celsius
A, B, C, D = Enthalpy Correlation Parameters

If a correlation is available in this form, with known values for the parameters A, B, C, and D, then the user may directly enter these parameters. If the user has a correlation of some other form, then the user should first perform curve fitting to obtain parameters according to the above correlation. Multiple enthalpy correlations for different temperature ranges may be used. Beyond the temperature range(s) of the enthalpy correlation(s) for a component, IDEAS will use the specific heat capacity to determine the enthalpy of the component.

Entering Proper Format Enthalpy Correlation Parameters:

If the enthalpy correlation parameters of the proper enthalpy correlation format (as described above) are known, they may be entered into the MP Assembler object directly.

1. Open the dialog box of the MP Assembler object.
2. On the "Quick Fits" tab, in the "Enthalpy" section, select the proper units (kJ/kg; kJ/kmole; or kcal/kmole) for enthalpy (specific enthalpy).
3. At the top of the Quick Fits tab, there is a Correlation File Attributes table. Below this table there is a field labeled "Next Empty Row". Copy the value shown in the "Next Empty Row" field to the field below the "Create in Row" button of the Enthalpy section of the "Quick Fits" tab.
4. Enter correlation parameters A, B, C, D, and the enthalpy of formation at 298.15 degrees Kelvin (25 degrees Celsius).

5. Enter the lower and upper limits of the applicable correlation range in the fields T_min and T_max. Multiple correlations may be added for enthalpy, with each correlation valid for its own temperature ranges. To add multiple enthalpy correlations, simply repeat this procedure for each correlation.
6. Press "Create in Row" button to save this correlation within the MP Assembler. Values will appear in the "Correlation File Attributes" table, indicating that the correlation has been created.
7. Proceed to "Saving Component Records" at the end of this section if no other correlations will be used for this component.

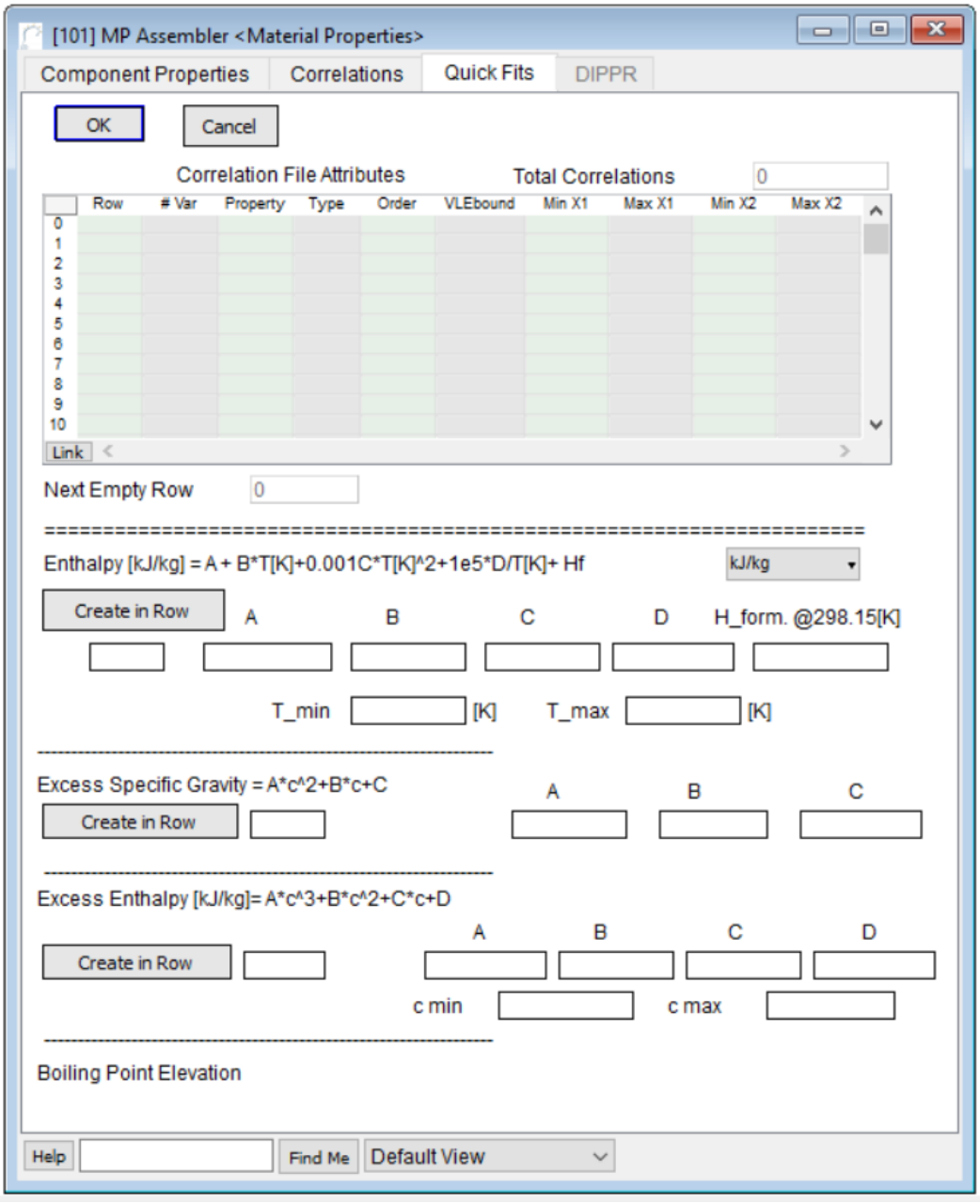


Figure 78: MP Assembler - Quick Fits - Enthalpy Correlation Created

E.4 Excess Specific Gravity (Volume of Mixing) Correlations

If no specific gravity correlation is specified, IDEAS will assume that there is no change in volume due to mixing of components. For many mixtures, this is a reasonable assumption. If specific gravity correlations are available, they may be entered into the component record file for better accuracy.

The correlation used within the IDEAS MP Assembler object Quick Fits tab for specific gravity is of the form:

SG = A*c^2 + B*c + C

Where: SG = Specific Gravity

c = Mass Fraction of Solute
A, B, C = Specific Gravity Correlation Parameters

1. Open the dialog box of the MP Assembler object, "Quick Fits" tab.
2. At the top of the Quick Fits tab there is a Correlation File Attributes table. Below this table there is a field labeled “Next Empty Row.” Copy the value shown in the “Next Empty Row” field to the field beside the "Create in Row" button of the Specific Gravity section of the “Quick Fits” tab.
3. Enter A, B, and C (the specific gravity correlation parameters).
4. In the Specific Gravity section, press the "Create in Row" button to save this correlation within the MP Assembler. Values will appear in the proper row of the “Correlation File Attributes” table, indicating that the correlation has been created (see Figure 79).
5. Proceed to “Saving Component Records” at the end of this document if no other correlations will be used for this component.

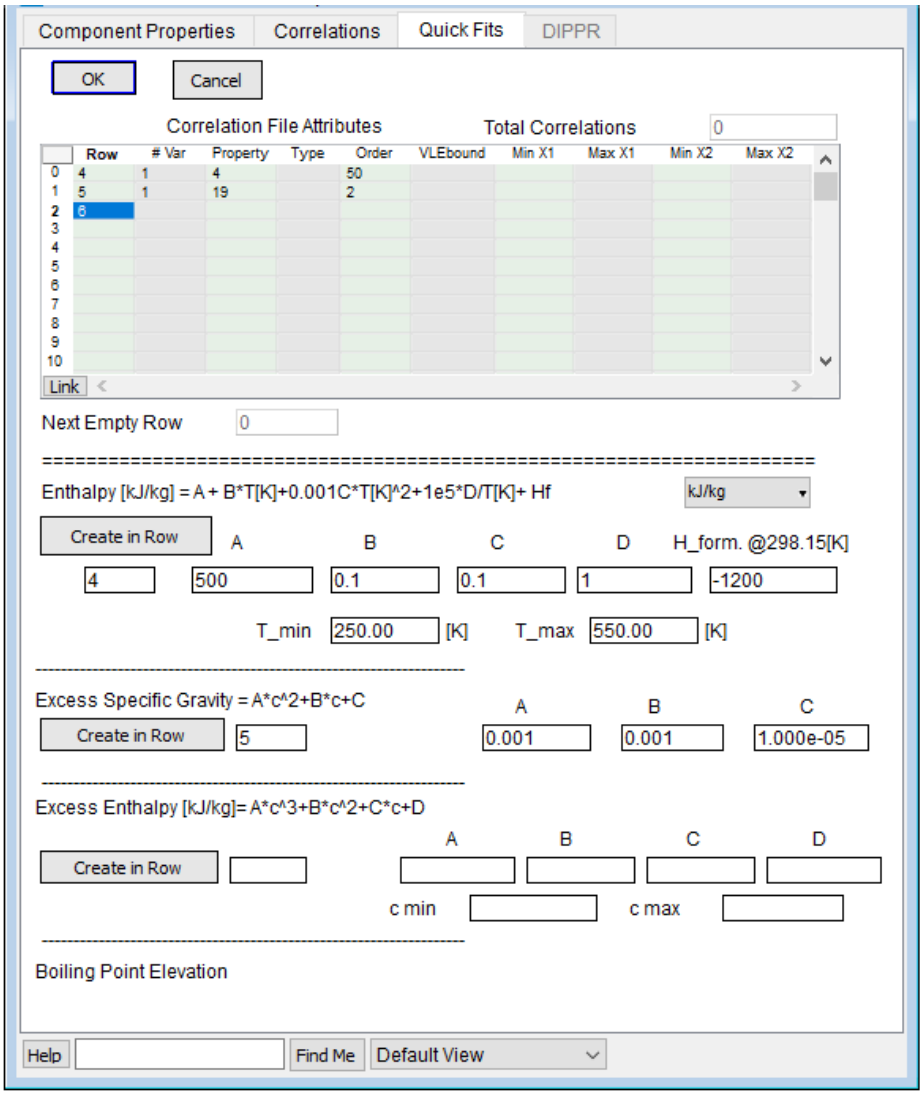


Figure 79: MP Assembler - Quick Fits – Specific Gravity Correlation Created

E.5 Excess Enthalpy (Heat of Mixing) Correlations

If no excess enthalpy correlation is specified, IDEAS will assume that there is no change in enthalpy due to mixing of components. For many mixtures, this is a reasonable assumption. If excess enthalpy correlations are available, for better accuracy, they may be entered into the component record file as described.

The correlation used within the IDEAS MP Assembler object Quick Fits tab for excess (specific) enthalpy is of the form:

excess_h = A*c^3 + B*c^2 + C*c + D

Where: excess_h = Excess Specific Enthalpy in kJ/kg

c = mass fraction

A, B, C, D = Excess Enthalpy Correlation Parameters

1. Open the dialog box of the MP Assembler object, "Quick Fits" tab.
2. At the top of the Quick Fits tab there is a Correlation File Attributes table. Below this table there is a field labeled “Next Empty Row.” Copy the value shown in the “Next Empty Row” field to the field beside the "Create in Row" button of the Excess Enthalpy section of the “Quick Fits” tab.
3. Enter the excess enthalpy correlation parameters: A, B, C, D.
4. Enter the lower and upper limits of the applicable correlation range in the fields c_min and c_max. See Figure 80. Multiple correlations may be added for excess enthalpy, with each correlation valid for its own temperature ranges. To add multiple enthalpy correlations, simply repeat this procedure for each correlation.
5. In the Excess Enthalpy section, press the "Create in Row" button to save this correlation within the MP Assembler. Values will appear in the proper row of the “Correlation File Attributes” table, indicating that the correlation has been created.
6. Proceed to “Saving Component Records” at the end of this document.

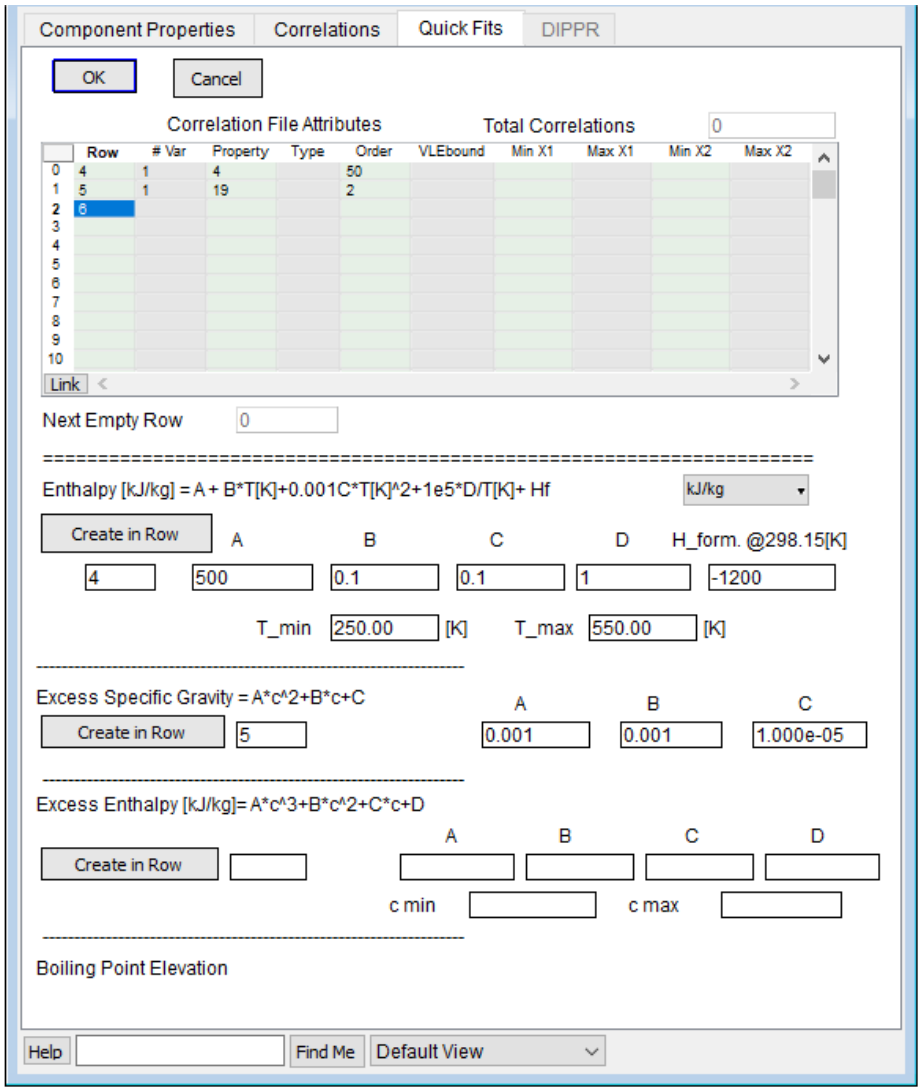


Figure 80: MP Assembler - Quick Fits – Excess Enthalpy – Ready to Create

E.6 Saving Component Records

In the MP Assembler object, Component Properties tab, press the "SAVE NEW RECORD" button to save the component in text file. Specify the name of the file, e.g. NewComponent.txt. Save the file as a text (.txt) file.

This component record is now available to be used within IDEAS models. As with all components, the component record must be loaded into the Material Properties object within an IDEAS worksheet, (“Components” tab, “Load Component” button) to be available for use in that worksheet.

The addendum describes the advanced feature of curve fitting to create coefficients for enthalpy and other

correlations.

Please contact the IDEAS Simulation Support team at simulation.support@andritz.com with any problems or questions.

E.7 Curve Fitting

Curve fitting is used to create correlations from known data. The IDEAS object “MP 1 Variable Curve Fitter” may be used to create correlation parameters for many physical properties. In this example, we will create correlation parameters for Enthalpy = F(T), enthalpy as a function of temperature. The object “MP 2 Variable Curve Fitter” is available to create correlations based on functions of two variables. See also the IDEAS help text for these two objects.

1. Place an MP 1 Variable Curve Fitter from the Material Properties library (MP Tools sub-menu) onto the IDEAS model. Open the dialog box.
2. Choose the radio button Enthalpy = F(T).
3. Enter data for enthalpy as a function of temperature into the Curve Fit Table on the Raw Data tab of the MP 1 Variable Curve Fitter object. Notice that the X column (temperature in degrees C in this case) is the right column, and Y (enthalpy in kJ/kg in this case) is the left column.
4. Press the "Compute Coefficients" button. IDEAS may announce that the order of the curve fit is one less than the number of data points provided. Click OK.

Note: On the "Curve Fit Status" tab, select the desired curve to be used (See Figure 81). The curve fit of the highest order available will provide the highest level of accuracy for the correlation data provided. However, for the remainder of the correlation range, a lower curve fit order correlation may be more accurate. The function Enthalpy(T) should always increase with temperature, but a high-order fitted curve may not meet this requirement. The button “Plot Y=f(X)” may be used to view the plot of each order correlation.

5. Select the desired curve to be used from the “Select This Curve For Database” area.
6. Press the "Save Fit in a File" button.
7. On the "Correlation Database" tab, press the "Save File to Database" button. Save the file with a representative name, such as NewCompEnthalpy.txt.
8. If desired, repeat this procedure to create additional enthalpy correlations for other temperature ranges of this component. It may be easier, however, to simply create one correlation using all known data for enthalpy as a function of temperature.
9. Open the dialog box of the MP Assembler object. Press the "Read Correlation File" button. Load the enthalpy correlation file that you saved in Step 8, for example, NewCompEnthalpy.txt. If additional enthalpy correlation files exist for this component, also load these correlation files by repeating this step.
10. Proceed to “Saving Component Records” at the end of this document if no other correlations will be used for this component.

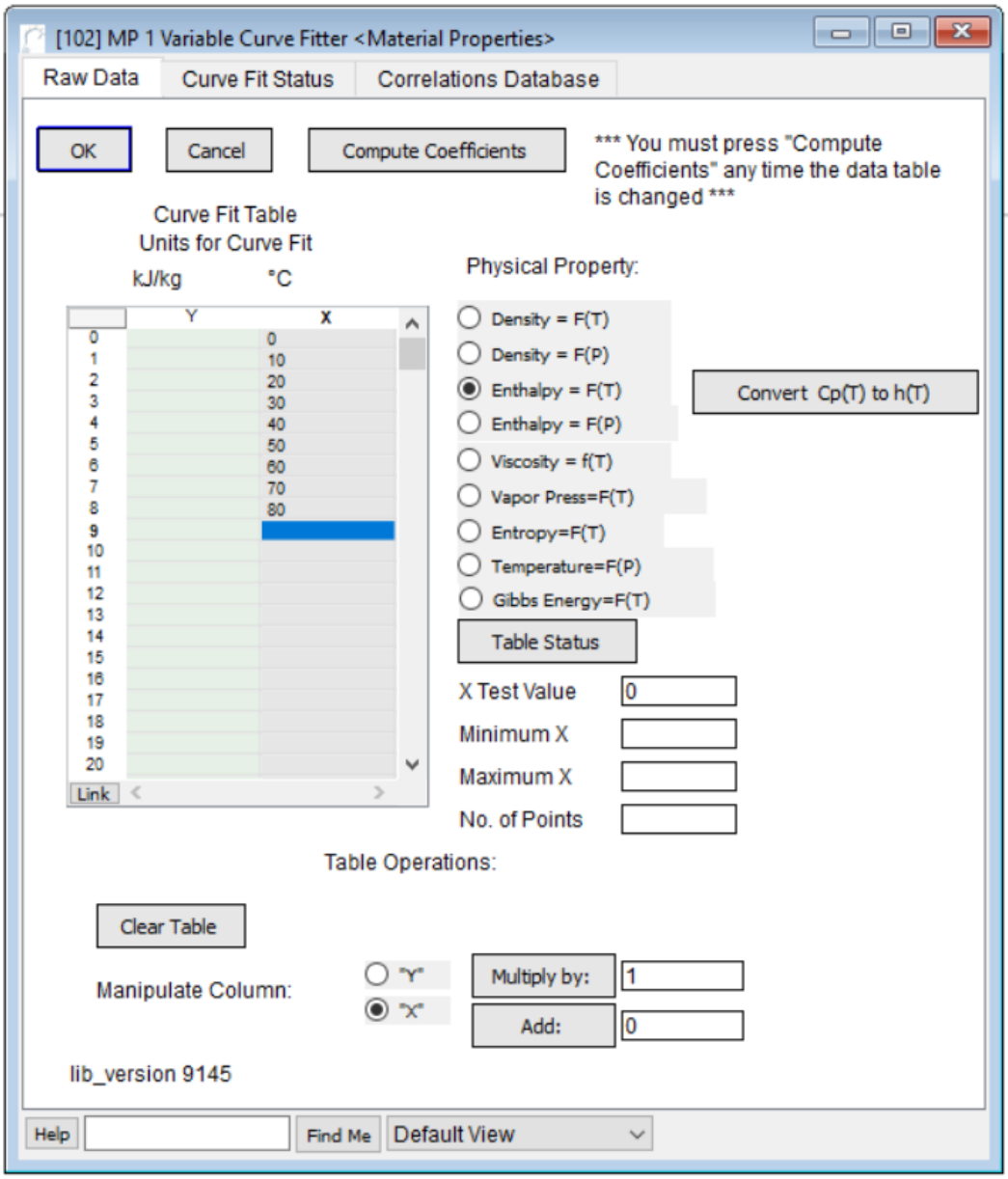


Figure 81: MP 1 Variable Curve Fitter – Raw Data – Coefficients Computed

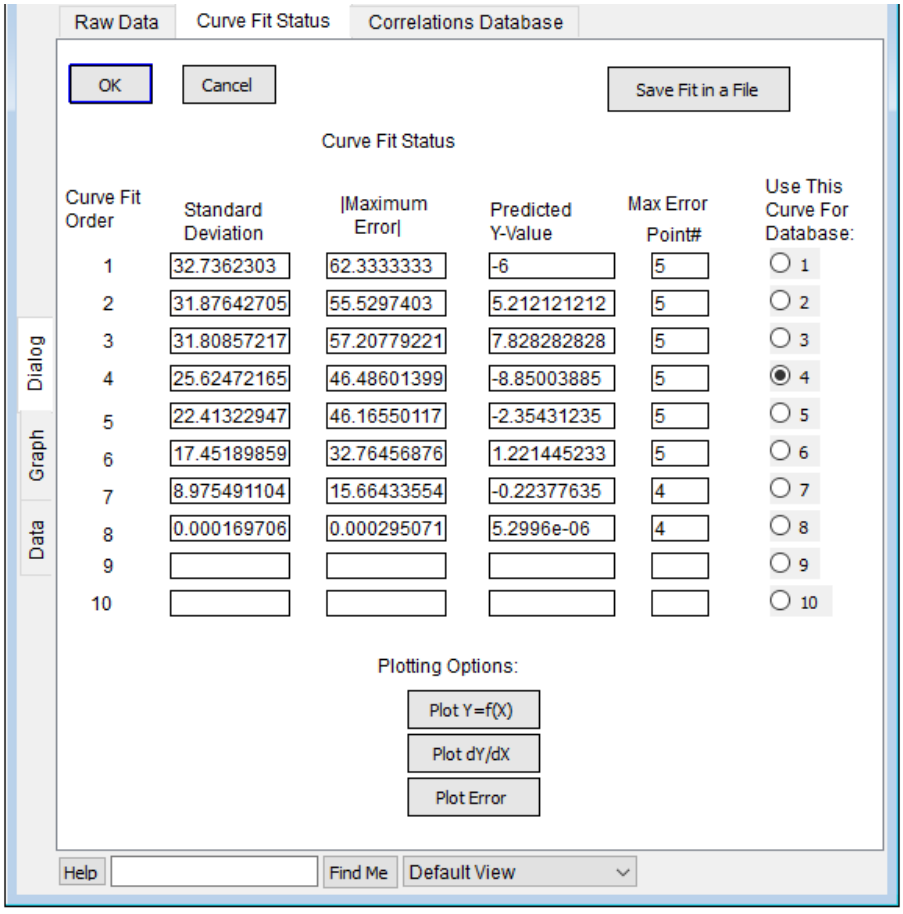


Figure 82: MP 1 Variable Curve Fitter – Curve Fit Status

Appendix F. Curve Fitting for Pumps and Fans

Digitizing the curve can be done by using the Graph Digitizer within the Tools_Utilities library.

F.1 Steps to be Followed for Using Graph Digitizer

1. Place the Graph Digitizer object from Tools_Utilities library onto the worksheet.
2. Double-click the object to open the dialog window.
3. Click on the browse button and select the file containing the curve you wish to digitize. Note: the file containing the curve must be converted into a bmp/jpeg file in order to digitize the curve.
4. After selecting the file, press Open. This pulls up the image from the file and a dialog box saying,
   "First you will be asked to define the axes and locate them on the image. You can redefine them at any time using the menu buttons. After defining the axes, click on the curve to register data points."
   Read carefully and press OK.
5. Define the value of the origin, x-axis maximum, and y-axis maximum. Click OK.
6. Another dialog box will appear saying,
   "Click on the (1) Origin, (2) X-max and (3) Y-max points in the image now. Only one click per item in the same order please!"
   Read carefully and press OK.
7. Click on the origin on the graph, next on end on the X axis (X-max) and finally on the end of the Y axis (Y-max). Immediately a message displaying "Next click on the curve to be digitized" appears. Press OK.
8. Now select data points of the curve by clicking the digitized graph. When you have selected all the required data points on the curve, click the green check mark (Accept and Close button) on the top of the toolbar.
9. The points that have been selected on the graph will be extracted and appear in X-Y data table of the Graph Digitizer dialog box.

For more information on using this object, refer to the help text.

The extracted data points from the Graph Digitizer can be used in the General Curve Fitter or Pump Curve Fitter to determine the pump curve coefficients.

F.2 Steps to be Followed for Using General Curve Fitter and Finding Coefficients of Curve

1. Place the object named General Curve Fitter, available in Tools_Utilities library, on the worksheet. After double-clicking on the object, the object's dialog box will open to the Input Data Table. Within the Data Table, enter a table of "Y" vs "X1" and/or "X2" data values. Then select between 1 Variable Function and 2 Variable Function radio buttons.
2. Click on the Step 1: Compute Coefficients button. The result can be seen in the Curve Fit Status table of Curve Fit Output tab.

Note: The maximum curve fit order is 10. If there are less than 11 data points are entered, you will get a message that says,“[w]ith only ‘n’ Y-X points the maximum order of the curve fit is ‘n-1,’” where ‘n’ is the total number of points. Press OK if this message appears.

3. Click on the Step 2: Show Coefficients button. A message will be displayed “Enter order of correlation valid orders are 1 to 10.” Now enter the value of order of the curve the user wants to fix in the box provided and press OK. The coefficients are computed and can be seen in Curve Fit Output tab.

The coefficients calculated using this General Curve Fitter can be used for fitting fan curves.

For more information on using this object, refer to the help text for the General Curve Fitter.

F.3 Steps to be Followed for Curve Fitting of Fans

1. Open the Pumps & Compressors library and place a Pump Curve Fitter object on to the worksheet. After double-clicking on the object, a dialog box will open to the Input Parameters tab. Select the appropriate Pump Type, Curve Type, Head, Flow, and Power units. Then enter the min and max values of X and Y axes in the Plotting Limits field.
2. Next in the Input Parameters tab, enter the Max. No of points that have been extracted from the graph/curve using the Graph Digitizer in the field provided.
3. Select Enter X, Y Data from the dropdown menu, which is besides the Enter Data button of Step 1. Press the Enter Data button.
4. A window where X, Y data has to be entered will open. Enter the X, Y data that is copied from the Graph Digitizer and paste it in the columns that are provided.
5. Press Fit Curve button. Immediately the coefficients (results) can be seen in Curve Fit Coefficients table of Curve Fits tab.
6. Copy the coefficients that are calculated by using Pump Curve Fitter object from the Curve Fits tab. Then paste them in the Pump Curves dP=f (Q)/Maximum Flow Curves; Fmax=f(rpm)/NPSH Curve; NPSH=f(Q) based on the curve that has to be fitted.

For more information, please refer to the Pump Curve Fitter help text.