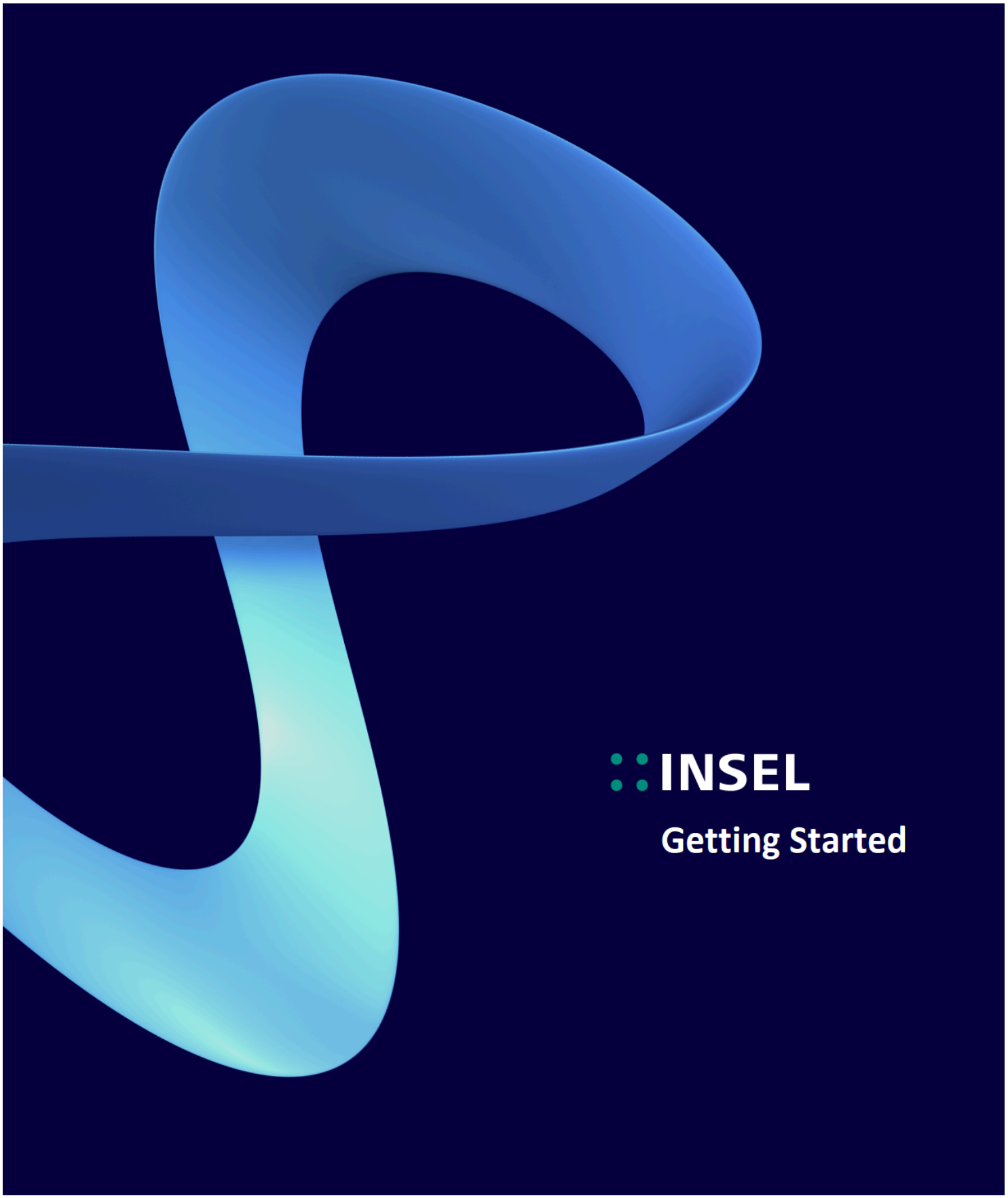

INSEL
Getting Started

INSEL 8 :: Getting Started



© 1986–2020 Jürgen Schumacher. All rights reserved.

Please visit us at www.insel.eu

Bison is free software and is available under the GNU General Public License.
Eclipse is a trademark of the Eclipse Foundation, Inc. and copyright protected by Eclipse contributors and others.
GCC is copyright protected by Free Software Foundation, Inc.
Flexx is copyright protected by The Regents of the University of California and The Flex Project.
gfortran is copyright protected by Free Software Foundation, Inc.
gnuplot Copyright 1986 – 1993, 1998, 2004 Thomas Williams, Colin Kelley
HP VEE is a registered trademark of Hewlett-Packard Co.
IISiBat is copyright protected (Centre Scientifique et Technique du Bâtiment CSTB).
INSEL was a registered trademark of doppelintegral GmbH, Stuttgart, Germany (liquidated in December 2019).
Current trademark/copyright holder to be confirmed.
Java is copyright protected by Sun Microsystems.
LabVIEW is a registered trademark of National Instruments Corporation.
Linux is a registered trademark of Linus Torvalds.
Mac OS X is a registered trademark by Apple, Inc. in the United States and other countries.
MATLAB is a registered trademark of The MathWorks Inc.
Microsoft Windows and Visual Studio is a registered trademark of Microsoft Corporation in the United States and other countries.
MiKTex-pdfTeX is copyright protected by D. E. Knuth and Han The Thanh
Ruby is copyright protected free software by Yukihiro Matsumoto
Simulink is a registered trademark of The MathWorks Inc.
Subversion is an open source version control system.
TeX is a trademark of the American Mathematical Society.
TRNSYS is copyright protected (S.A. Klein, F.L Alvarado).
VSEit is copyright protected by Kai Brassel.
Window Builder is provided under the terms and conditions of the Eclipse Foundation Software User Agreement.

About this manual

This manual is thought as a first introduction to version INSEL 8 of the graphical programming language INSEL. No knowledge about older versions or even the concept of graphical programming is required.

The older versions INSEL 5, 6, and 7 made extensive use of the graphical programming environment HP VEE Runtime by the Hewlett-Packard company. The interface of INSEL 8 is now completely based on VSEit, the Versatile Simulation Environment for the Internet (pronounced as use-it) by Kai Brassel. VSEit is a Java-based framework and therefore platform independent. In order to avoid confusion, this manual speaks of INSEL features although most of the graphical concepts are in fact VSEit features.

INSEL 8

The INSEL 8 software can be downloaded under www.insel.eu in the Download area. It can be installed on a Microsoft Windows computer (32 or 64 bit), or under macOS (64 Bit). Several Linux distributions (32 or 64 bit) are available, too.

License agreement

A license agreement must be confirmed before the INSEL 8 software can be installed. All the terms in this license agreement apply to this document, too. The license agreement can be read and printed before the installation of INSEL 8 starts.

1 :: Quick start

INSEL is an acronym for Integrated Simulation Environment Language, a graphical programming language for the creation and execution of simulation applications mainly used in the renewable energy sector.

This chapter starts with some installation hints and shows the basic ideas how to use the graphical user interface of INSEL can be used to create programs and execute them.

1.1 Installation

The available INSEL 8 installers for supported operating system are

- `win32\setup_insel_8.3.0_32.exe` (Windows 10)
- `win64\setup_insel_8.3.0_64.exe` (Windows 10)
- `macOSX/insel_8.3.0_macOS.pkg` (macOS Mojave)
- `linux32/insel_8.3.0_32.deb` (Debian Linux)
- `linux64/insel_8.3.0_64.deb` (Debian Linux)
- `linux32/insel_8.3.0_32.rpm` (Red Hat Linux)
- `linux64/insel_8.3.0_64.rpm` (Red Hat Linux)

Administrator rights required

INSEL 8.3 will be the last version to support 32-bit operating systems. The setup program requires administrator rights in order to install the software.

1.1.1 Windows

All files, executables and dynamic link libraries which are required by INSEL 8 are copied to the INSEL installation directory, which is typically

`C:\Program Files\INSEL 8.3` or `C:\Program Files (x86)\INSEL 8.3`

when the 32-bit version is installed under 64-bit Windows. The directory includes a copy of the Java Development Kit JDK Version 8 since the VSEit user-interface of INSEL is based on Java 8.

The `resources` directory of the INSEL installation is added to the Windows environment variable `%PATH%`.

Documentation

The INSEL documentation is written in LaTeX and is supplied in .pdf format. The Windows version uses Adobe's Acrobat Reader. In case, no Acrobat Reader can be found a fallback to the reader provided in the `resources` directory is used. In most Debian distributions Document Viewer is installed by default.

1.1.2 macOS

The `.pkg` package can be installed by a double click on its icon. The directory structure is similar to the Windows version. The application is typically installed as `/Applications/INSEL.app` macOS application bundle.

Symbolic links

Symbolic links are created to all dynamic libraries and three executables which INSEL uses. A complete list can be found in the shell script `INSEL.app/Contents/_createRequiredSymbolicLinks.sh`.

GNUPLOT GOES HERE: SEE PROGRAMMERS GUIDE (TEMPORARILY)

1.2 Starting and ending INSEL 8



In the Windows version the installation program has created an icon on your desktop. INSEL 8 can simply be started with a double click on the icon.

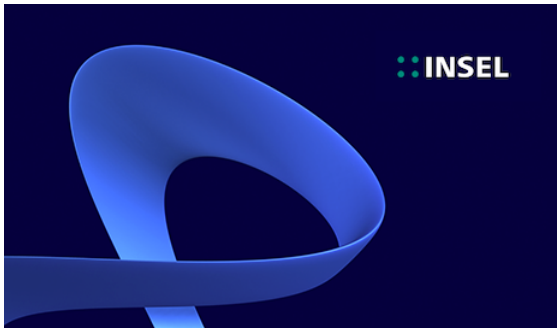
Another possibility is to start INSEL 8 via Windows' Start button and the link to the executable in the Programs list (group `insel 8`).

Yet another possibility is to browse to the installation directory (typically `C:\Program Files\INSEL 8.3`) with the Windows Explorer and double-click on the executable `insel_8.exe`.

There is another executable named `insel.exe` in the `resources` subdirectory. This executable is meant to run INSEL 8 from a console prompt or in batch mode.

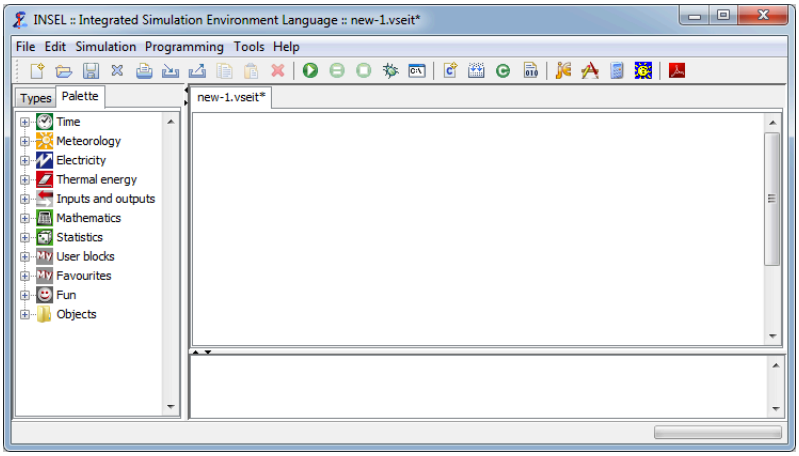
Splash screen

During start INSEL 8 will display a splash screen.



INSEL window

After a moment the INSEL window appears.



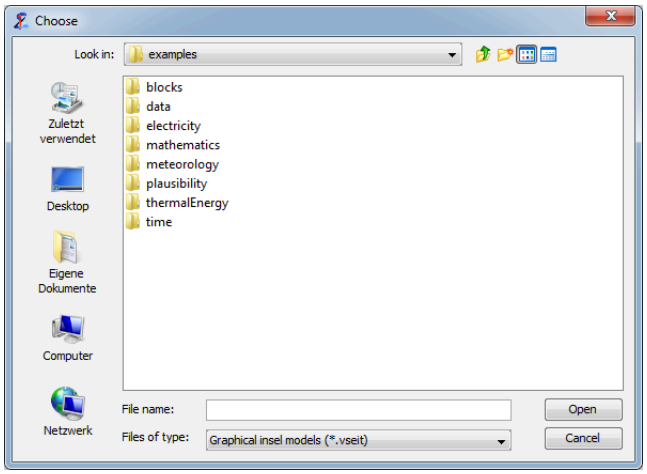
- The *Title* bar contains the program name, the name of the currently open document (like `new-1.vseit` for the first open, empty model), and the standard Windows buttons to minimize, maximize and close the window.
- The *Menu* bar with the *File*, *Edit*, *Simulation*, *Programming*, *Tools*, and *Help* menu can be used to access items and features.
- The *Tool* bar and its icons provide buttons for the most frequently used functions.
- On the left-hand side the *Palette* is displayed. Here all INSEL blocks can be found. They are organized by *categories*, like *Time*, *Meteorology*, *Electricity* etc.
- Next to the palette is the *Types* tab. Similar to the palette the *Types* pane lists INSEL blocks, but only those used in the current model.
- The white space on the right side is the *Work area* which can be used to create INSEL block diagrams.
- The *Output window* below the work area is used by INSEL for text output.
- The *Status* bar is displayed at the bottom of the INSEL window. It is used for temporary text messages and includes the *Progress* bar which shows the progress of a running INSEL model.

Ending INSEL 8

The INSEL main window can be moved, resized and closed in the usual fashion. The *File – Exit...* menu item and the *Close* button in the window's title bar are equivalent options to end the program.

1.3 Running a first example

A good starting point to become familiar with INSEL are the examples which can be found in the `examples` directory. The easiest way to access the examples is via the *File – Open example...* menu item which will open a file chooser dialog similar to the one in the next figure.

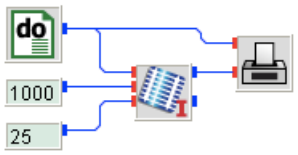


`blocks` directory

In the `blocks` directory one basic example for each INSEL block is available. It is certainly a good idea to browse through the subdirectories of the `blocks` directory and get an overview which blocks are available in INSEL 8 and what their functions are.

`pvi.vseit`

As a first example we choose `pvi.vseit` from the `electricity` directory. The block diagram looks like this:

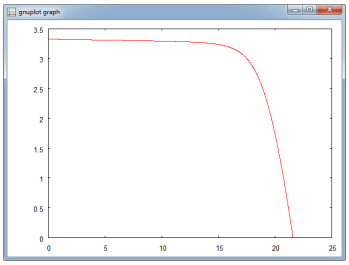


The application will plot the I-V curve of a PV module under standard test conditions. Before going into details let us execute the model.



In the tool bar there are five buttons dealing with the execution of INSEL models.

Since `pvi.vseit` is the current not-yet-running INSEL model the *Run* button is highlighted. Your first INSEL simulation run is now only one mouse click away. The result should be a Gnuplot graph.



Should it disturb you that Gnuplot displays mouse coordinates in the lower left corner by default, press the *m* key and they disappear. Pressing the *m* key again brings them back.

Debugging

Another option to execute INSEL models is via the *Debug* button. In this case the block which has the focus is highlighted by a green frame. The current values of all inputs and outputs are shown next to the respective ports. In addition, INSEL will display a list in the output window which block is called at the moment.

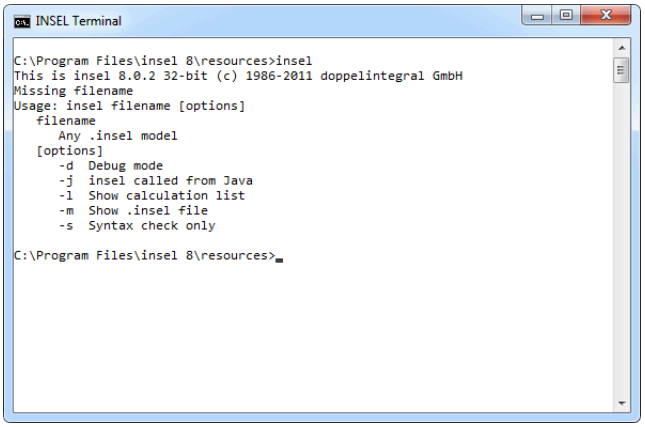
When you try out the debug mode, you can observe

- How much slower the execution of the model is.
- How the progress bar indicates the status of the running INSEL model.
- That the *Pause* button and the *Stop* button are highlighted during model execution.

Obviously, a click on the *Pause* button pauses model execution and a econd click on the *Pause* button continues model execution. The *Stop* button terminates model execution. However, INSEL prompts you to confirm this action.

Run INSEL from DOS prompt

It is also possible to execute INSEL models in batch mode or in a terminal or DOS box window. Such a terminal or DOS box window can be opened via the rightmost button C:\, for example. At this point it is sufficient to have a look at the usage of the `insel` command.



More information on this topic can be found in section *INSEL without GUI* of the Tutorial's Module 13.

1.4 INSEL blocks in VSEit

As mentioned before, the graphical user interface of INSEL 8 – the INSEL window – is based on the VSEit framework. Before we take a closer look at the usage of INSEL blocks in VSEit, a basic understanding of the term INSEL block is required.

Question

So let us ask the question "What is an INSEL block?" and try to answer it. Well, in principal, an INSEL block is nothing but a representation of a mathematical function, let's say f.

Inputs and outputs

Most functions depend on independent variables $x = (x_1, x_2, \ldots)$. In INSEL, these independent variables are called inputs. When the function f is applied to x, the resulting dependent variables $y = (y_1, y_2, \ldots)$ are called outputs. Please notice, that the x_i and y_i are scalars (4-byte real numbers in INSEL).

Hence, we may write $y = f(x)$, and interpret this as a representation of an INSEL block named f with inputs x and outputs y.

Parameters

In addition, the function – or INSEL block – f can have a set of constant parameters $p = p_1, p_2, \ldots)$ which influence the current value of the output y. Hence, we can write $y = f(x, p)$. Parameters in INSEL can be real numbers and strings.

History

Finally, blocks may have a history, which can make their results time-dependent, i. e., every INSEL block can be represented by the equation

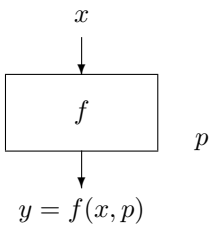
$$y = f(x, p, t)$$

Answer 1

In conclusion, it follows from these remarks that an INSEL block is a representation of

an explicit function, i. e., $y = f(x, y, p, t)$ is not allowed.[1]

Answer 2 Because it's a graphical programming language, INSEL represents the equation $y = f(x, p, t)$ by a graphical element. This picture is called INSEL block, too:



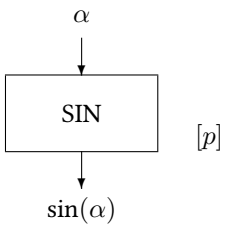
In the documentation, a rectangle is used to represent an INSEL block named f, for example. Inputs x come into the block via arrows which point into the rectangle from the top. Outputs y are represented by arrows pointing out of the block.

It is convenient to write the names of the inputs and outputs close to the corresponding arrows. Parameters are frequently written at the right edge of a block. As mentioned before, the number of inputs, outputs and parameters can range from zero to any positive number and depend only on the specific requirements of the block f.

By convention, block names in INSEL use all capital letters.

SIN block In the simple case of the function $y = \sin(\alpha)$ the block representation looks like





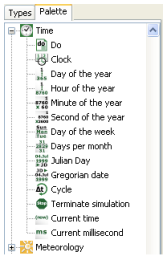
The SIN block requires exactly one input, namely an angle α. It returns exactly one output, the sine of α. The parameter $[p]$ is written in square brackets, which means that the parameter p is optional, i. e., not necessarily required.

Angles can be given in either degrees or radians. How does the SIN block know what is meant? By default, the SIN block assumes that α is given in degrees. This is equivalent to not specifying p at all, or by setting the parameter p equal to zero. If you want the SIN block to recognize α in radians you have to set p equal to 1. Any other value for p will result in an error message.

[1] There are exceptions to this rule in INSEL, but this is not the place to discuss them.

1.4.1 The Palette

The most convenient way to access and use INSEL blocks is from the Palette.

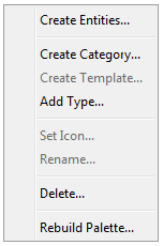


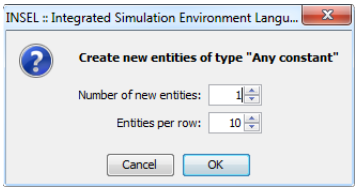
A category (like *Time*, for instance) can be opened by a left-mouse click on the small + symbol next to the icon of the category. An open category can be closed by a left-mouse click on the small − symbol. The opened *Time* category with its types is shown in the margin.

It contains the DO block (type *Do*), the CLOCK block (type *Clock*), the DOY block (type *Day of the year*) and so on.

There are three ways to insert INSEL blocks—or more precisely, entities of the block—into an INSEL model.

- A block[2] can be dragged from the palette into the work area by keeping the left mouse button pressed. INSEL displays a small + sign next to the mouse pointer once the mouse is moved into the work area. When the mouse button is released an entity of the block will be placed in the work area.
- When a block (i. e., type) is selected from the palette by a left-mouse click, then a click in the work area places a new entity of the selected block where you like. The selection of the block in the palette disappears. More than one entity of a marked block can be inserted in the work area as long as the *Shift* key on the keyboard is kept pressed. In this case, each mouse click in the work area will place one entity of the block in the work area.
- In order to create several copies of a block at once, select a type, the context menu of the palette can be opened by a right-mouse click in the palette area, and choose *Create Entities...*. In a dialog the number of required entities can be specified.





The new entities will be placed in the work area, arranged according to the number of entities per row.

Customizing the palette

[2] It would be more precise to say type, but in many cases it is more convenient to simply speak of blocks when the context is clear enough.

In INSEL 8 each user of a computer has an own copy of the palette. The location, where the user palette is stored can be on the local machine or on a remote computer, depending on the user-profile settings.

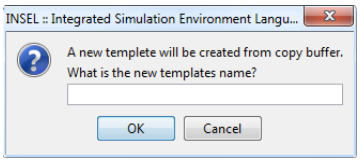
Palettes can be fully customized. Categories and types can be dragged with a left-mouse click to a different position within the palette. Keeping the *ctrl* key pressed (*cmd* on a Mac keyboard) creates a copy of the selected item instead of moving it.

New categories

The creation of new categories should be obvious. The position of the new category depends on the current selection within the palette. If nothing is selected, the new category will be created at the root of the palette.

New templates

All other palette operations can be made via the context menu, shown above. When one or more entities are selected in the work area, the *Create Template...* item is enabled. Choosing it, opens a dialog in which the name for the new template can be specified.



Please notice, that a template can contain more than just one block. Therefore, a template is not the same as a type.

New types

The creation of new types is a very advanced option and is explained in Module 11 :: *Programming INSEL blocks* of the Tutorial.

Etc.

All user-defined categories, templates, and types can have user-defined icons, they can be renamed and deleted at any time.

Rebuild palette

The factory setting of the palette can be rebuild. All user-made changes will be saved in a category named *SAVED ENTRIES*. The rebuild process is initiated only after a confirmation dialog.

1.4.2 Block entities

When INSEL blocks (entities) are created from the palette, they appear in the work area as minimised icons. The following picture shows a DO block and a SCREEN block, taken from the *Time* and the *Inputs and Outputs* category, respectively.



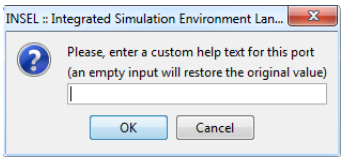
By default, each INSEL block is represented by a 32 times 32 pixel icon encapsulated in a frame. Block inputs are displayed at the left border, block outputs at the right side of the block's symbol.

Port tooltips

Block inputs and outputs can be accessed via ports. All ports in INSEL 8 have a tooltip

which displays helpful information to the meaning of the port. When the mouse pointer is moved to a port, the tooltip is shown after a short moment.

Default port tooltips can be overwritten. A double-click on a port opens a dialog where individual tooltips can be specified.



Connecting blocks

In order to connect an output port with an input port, click near one of the ports you wish to connect, keep the mouse button pressed and move the mouse pointer to the other port to be connected. Once you are close enough a small rectangle will show up, indicating that a release of the mouse button connects the selected ports.



Once connected, a connection line between input and output is shown. In general, an input port can only be connected with exactly one output port while output ports can be connected to an arbitrary number of different input ports.

Deleting connections

Block connections can be deleted one by one. A mouse click somewhere on the route of the connection and choosing *Delete* from the *Edit* menu or pressing the *Delete* key will dissolve the connection. When a block is deleted all its connections will be deleted automatically.

Trouble

If plenty of connection routes exist in a larger INSEL model, the routing algorithm – that is the part of VSEit which tries to find an ideal route for all connection lines – may not be able to find such a route and uses a fall back. In this case a short diagonal connection will be displayed which cannot be clicked on. If this happens, try to move the respective block and find a better routing for the block and then delete the connection.

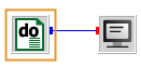
Moving blocks

Blocks can be moved to a different position in the work area by dragging them with pressed mouse button. The work area itself is infinite, which means, if you drag a block out of the visible part of the work area scroll bars will appear automatically.

More than one block can be moved at a time by selecting them first.

Selecting and deselecting blocks

A block can be selected by a mouse click on its icon. A frame will appear, indicating that a block is currently selected.



More than one block can be selected with the *Shift* key pressed. When more than one block is selected, a click any block deselects all other blocks, their frames disappear.

Alternatively, blocks can be caught with a rectangle that is created in the work area with the mouse, starting in the upper left corner and dragging the mouse pointer to a location in the lower right of the starting point.

All blocks can be selected using the *Select All* context menu.

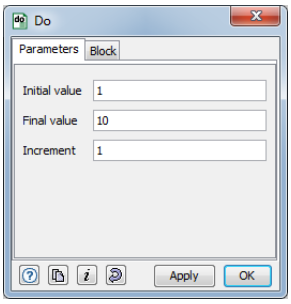
All selected blocks can be deselected by a mouse click on the white space of the work area. The selection of individual blocks can be inverted by pressing the *Ctrl* key (or *cmd* on Mac).

Deleting blocks

Selected blocks can be deleted by pressing the *Delete* key or via the *Edit – Delete* menu item or via a click on the Delete button in the tool bar. Please notice, that this action cannot be undone in the current version INSEL 8.3.

1.4.3 Entity editors

A double click on any INSEL block in the work area opens its entity editor. The range of entity editors varies from trivial to rather complex. This is the entity editor of the DO block:



Most INSEL blocks (better: entities) have two tabs in common: a *Parameters* tab and a *Block* tab.

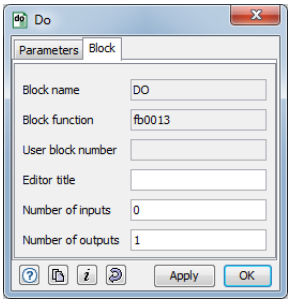
Parameters pane

The most important feature of the parameters pane is that you can access and modify all parameters of the coresponding INSEL block here. In case of the DO block these are the *Inital value*, *Final value*, and *Increment* parameters.

It is possible to use names of global variables (created with the DEFCON block) in any of the parameter fields. In the shown example, the DO block simply counts from one to ten.

Block pane

A click on the *Block* tab displays the block pane.



The block pane of an entity editor shows some insight information for the INSEL block, like the "real" *Block name* (DO, in this case), the name of the exported Fortran subroutine or C function implemented in a dynamic library, and the so-called *User block number* u, fixed by the inselEngine during model compilation.

Editor title

For each INSEL block entity a user-defined *Editor title* can be provided. The text will appear in the title bar of the entity editor and as a roll-over tooltip when the mouse pointer is moved across the minimized block in the block diagram. If no editor title is specified the roll-over tooltip displays the type name from the palette (e. g., Do), by default. After the first compilation of the model the default roll-over tooltip will change to type: u.

Input/output interface

The *Number of inputs* and the *Number of outputs* can be specified in the block pane of the entity editor – within block-specific limits for the number of allowed inputs and outputs.



The buttons in the bottom of the entity editor can be used for the following purposes (from left to right).

- The *Help* button gives direct access to the INSEL block reference page (Prerequisite: Adobe Reader must be installed).
- The *Clone* button opens another copy of the current editor. Changes made to any copy of an editor are synchronised whenever the *Apply* or the *OK* button is clicked.
- The *Info* button toggles the display of information to the block parameters, usually the physical units of the corresponding parameters.
- The *Reset* button rejects all unsaved changes made to the parameter settings and resets all attributes to the recently stored values.
- The *Apply* button saves the current values of all attributes without closing the entity editor window.
- The *OK* button saves the current values of all attributes and closes the entity editor window.

Execution

Well, when the model consisting of the DO block and the SCREEN block as used in the

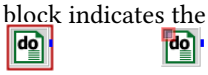
discussion so far, is executed the result is displayed in the output window.

```
Compiling new-1.vseit ...
No errors or warnings
Running insel 8.3.0 ...
   1.0000000
   2.0000000
   3.0000000
Normal end of run
```

Not very spectacular, but this example shows that everything seems to work.

1.4.4 Errors in networks

When a model has syntax errors like inconsistent number of inputs or parameters, for example, a red frame around the block indicates the block which causes the problem.

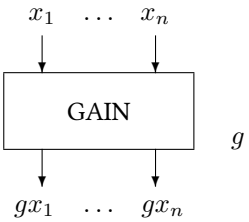


In addition, a toogle button in the *Types* pane can be used to indicate incomplete types and entities. If *on*, a small red square is shown in the upper left corner of entities which are causing problems.

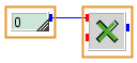
1.5 Macros

As mentioned above, a collection of INSEL block entities can be saved in the palette as a template for further use. If, for example, we would like to rebuild the GAIN block





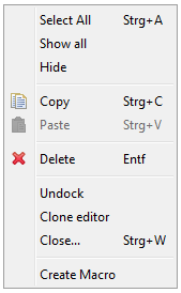
from a CONST block – type *Any constant*, and a MUL block – type *Multiplication*.



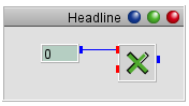
we'd select both blocks and create a template via the palette's context menu.

Creating and dissolving macros

However, it might be preferable, to combine both blocks into a macro and save the macro as template. We can do so by using the *Edit – Create Macro* menu or by using the context menu with a right-mouse click on the empty work area.



This will create a macro with a default title bar.



The title bar shows three buttons. From left to right, they can be used to open the entity editor of the macro, to maximize the macro to the full size of the current work area, and reduce its size again. The right button minimizes the macro. All other properties of the macro are inherited from ordinary entities.

Please observe that it is not necessary to add a macro input and a macro output port for the MUL block, because inner ports in the macro can be connected to other blocks across the border of the macro.



Only if you wish to be able to connect to the MUL block even when the macro is minimized the corresponding ports should be created via the macro's *Edit* function and its blocks pane. In this case, the minimized macro looks like any other INSEL block, as shown in the margin.

An option to create empty macros from scratch is to use the *Macro* type from the palette's *User blocks* category.

Macros can be created within macros. There is practically no limit for the depth of nested macros in INSEL.

One or more selected macros can be dissolved via the *Edit – Dissolve macros* menu or the macro's context menu.

Editing macros

In addition to using the buttons in the macro's title bar, the size of an opened macro can be modified by dragging its lower right corner with the mouse.

Macros can be moved by picking them up at the title bar or their frame, as indicated by a hand symbol of the mouse pointer.

Blocks can be moved from the work area into a macro by dragging them to the macro area. This is possible only if the target macro is opened. The drop option is indicated by a frame around the target macro. Please observe that any previous port connections will be conserved and that the ports are adapted accordingly. The same applies, when blocks are dragged out of a macro.

und sie bewegen sich doch (die Ports).

mit rev. 868 sollte nun das Ändern der Reihenfolge von Makro-Ports möglich sein. Dazu muss der Benutzer nur CMD/CTRL drücken, den Port anfassen und vertikal verschieben. Statt CMD/CTRL geht auch Klicken und Halten auf dem Port bis der Curser wechselt. Achtung: Beim Editieren der Makro-Hierarchie gibt es Situationen, in denen das Programm die Reihenfolge der Ports selbst wieder neu bestimmt.

Mit rev. 869 verschwinden Ports nicht mehr, wenn der letzte innere Link verschwindet. Achtung: This way, "orphan" ports may be created. These can only be deleted by shifting them to the bottom of the makro and, then, set the number of ports to a suited value.