



# EcosimPro Version 4.8 Upgrade Presentation

**EA Internacional**  
**8th September 2010**

**Phone: 34 – 91 448 85 98**

**http: [www.ecosimpro.com](http://www.ecosimpro.com)**

**E-mail: [info@ecosimpro.com](mailto:info@ecosimpro.com)**



**This presentation shows some of the new capabilities in EcosimPro 4.8**

# Summary



- **General**
- **Management**
- **EL language**
- **Monitor**
- **Experiment Wizards**



# General Improvements

# Main improvements



- **Apart from Microsoft C++ compilers, from version 4.8 it can be used GNU C++ compiler. This new compiler is free and it will be included in the default installation.**
- **New widget tool for monitoring the simulation over a schematic in the Monitor tool. The widget includes alarms checking for detecting out-of-limits values**
- **New capabilities in plotters for using two independent Y-axis**
- **Markers in plotters to get exact coordinates**
- **New tool for exporting monitor tables in CSV format**
- **New experiment wizards for setting data, functions, etc.**

# Valid C++ compilers

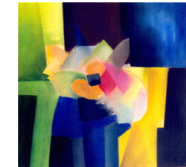


- **Simulation phase:**
  - **Valid compilers for using EcosimPro 4.8:**
    - MS Visual C++ 6.0
    - MS Visual Studio C++ 2003
    - MS Visual Studio C++ 2008
    - GCC 4.4.0 Windows (**NEW**)
- **Deck generation (runtime models) from EcosimPro models:**
  - All previous MS compilers under Windows
  - Windows & GCC 4.4.0



# Management Improvements

# Managing Compilers

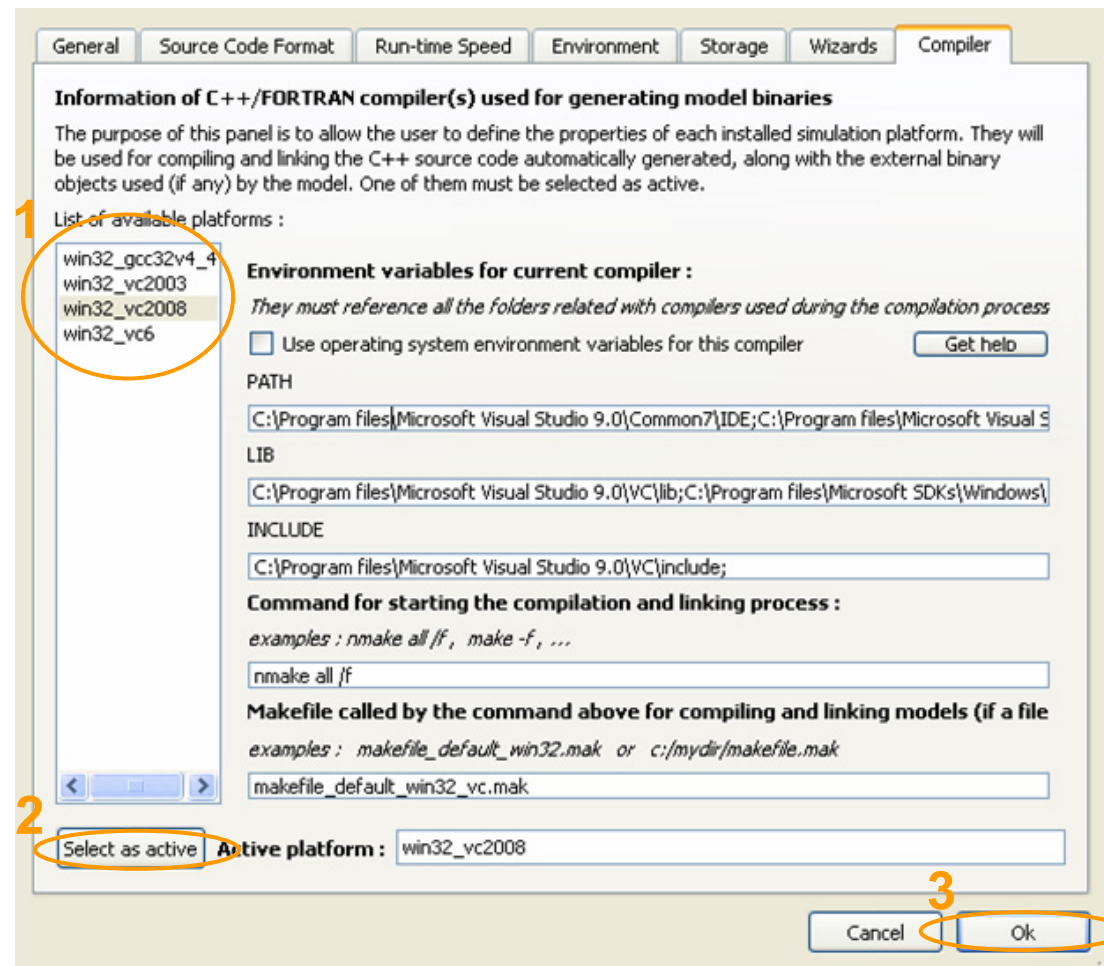


The user can decide within the program using the panel Preference Options the active platform to compile and run models

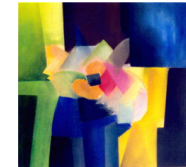
Now using this panel the user can manage the set of values related with some locations required by the compiler within EcosimPro by editing the fields PATH, LIB and INCLUDE, so it is not necessary anymore to set any environment variable externally

Steps:

- 1.- Select compiler
- 2.- Mark as active
- 3.- Accept



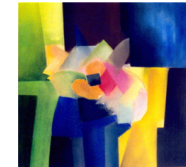
# External Objects Organization regarding compilers



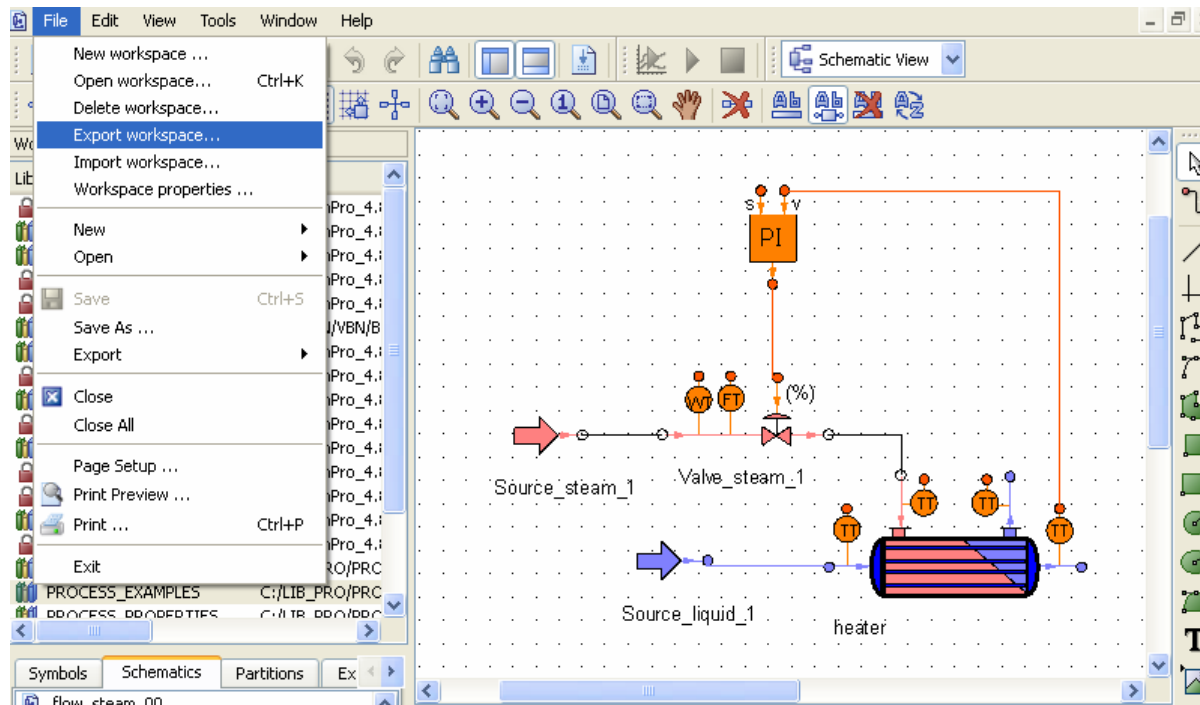
- **At EcosimPro modelling library level, the external binary objects used by models can be located by default in two places:**
  - **Inside the /lib/[PLATFORM]/ folder of each modelling library, using its own name in lower case and without any suffix for identifying the platform.**
  - **Inside the /lib folder of each modelling library using the convention of names name\_plattform.extension and using lower-case.**

PLATFORM	USAGE	EXTERNAL OBJECT INSIDE A SUBFOLDER WITH PLATFORM INFORMATION	EXTERNAL OBJECT LOCATED DIRECTLY INSIDE /LIB FOLDER
win32_vc6	MODELS AND DECKS	/lib/win32_vc6/turbo.lib	/lib/turbo_win32_vc6.lib
win32_vc2003	MODELS AND DECKS	/lib/win32_vc2003/turbo.lib	/lib/turbo_win32_vc2003.lib
win32_vc2008	MODELS AND DECKS	/lib/win32_vc2008/turbo.lib	/lib/turbo_win32_vc2008.lib
win32_gcc	ONLY DECKS	/lib/win32_gcc/turbo.a	/lib/turbo_win32_gcc.a
win32_gcc32v4_4	MODELS AND DECKS	/lib/win32_gcc32v4_4/turbo.a	/lib/turbo_win32_gcc32v4_4.a
linux_gcc	ONLY DECKS	/lib/linux_gcc/turbo.a	/lib/turbo_linux_gcc.a
linux_gcc32v4_4	ONLY DECKS	/lib/linux_gcc32v4_4/turbo.a	/lib/turbo_linux_gcc32v4_4.a

# Export/Import Workspace

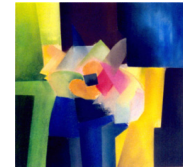


**EXPORT:** It saves the configuration of the workspace to export. This configuration file contains the libraries name of the exported workspace and will be saved with file extension “.wsp.xml”



**IMPORT:** Once the workspace file has been selected, the libraries and their components will be displayed in the workspace area

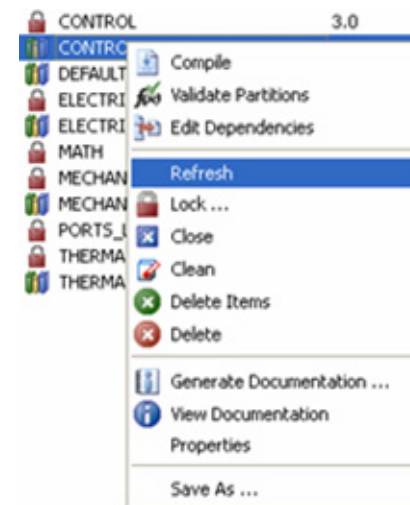
# Multiuser File Protection System

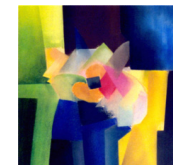


- New software package for EcosimPro, which aim is to avoid potentially dangerous situations when two or more users are working simultaneously with a shared library, by means of warning messages and blocking actions

- Schematic protection (.eds files)
- Experiment files operation (.exp files)
- Partition protection
- Integrity Protection of libraries

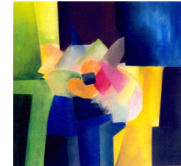
- Furthermore, the user can refresh the status of the libraries with the “refresh” option in the library contextual menu





# EL Improvements

# getExperimentDir() FUNCTION



- The getExperimentDir() function returns the original experiment path. If the experiment directory has not been changed, this function returns the same value as getWorkingDir()

The syntax is as follows:

```
STRING getExperimentDir()
```

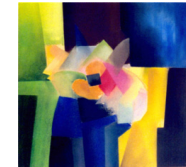
- **Example:**

```
EXPERIMENT exp1 ON compo1.default
DECLS
    STRING dir
    STRING inputFile= "myInput.txt"
BODY
    dir= getWorkingDir()
    WRITE("Actual working directory: %s\n",dir)
    dir= getExperimentDir()
    WRITE("Original experiment directory: %s\n",dir)
    WRITES(myInputFile,"%s%s",getExperimentDir(),myInputFile)
    WRITE("Input file: %s\n",myInputFile)
END EXPERIMENT
```

- **Printed message:**

```
Actual working directory: D:/myOputups/USR1259/DEFAULT_LIB.compo1.default.exp1.2010_06_13_11_22_15/
Original working directory: C:/InstallDir/USER_LIBS/experiments/compo1.default/exp1/
Input file: C:/InstallDir/USER_LIBS/experiments/compo1.default/exp1/myInput.txt
```

# Additional RESET() Actions



- The `setCallAfterReset( functionPtr )` function should be used when it is necessary to execute some actions contained in the "functionPtr" function after every call to the `RESET_VARIABLES()`. Typically it is used to change default data to new data, restore values from a file, read values from a database, etc.

The syntax is as follows:

```
NO_TYPE setCallAfterReset(FUNC_PTR functionPtr )
```

- Example:

```
FUNCTION NO_TYPE setInitialValues()
BODY
    a= 63.77
END FUNCTION
EXPERIMENT exp1 ON compo1.default
DECLS
    STRING dir
    STRING inputFile= "myInput.txt"
BODY
    -- print "a" after resetting default values
    RESET_VARIABLES()
    WRITE("Before a= %g\n",a)
    -- activates the function
    setCallAfterReset(setInitialValues)
    RESET_VARIABLES()
    WRITE("After a= %g\n",a)
    -- deactivate the function
    ussetCallAfterReset(setInitialValues)
END EXPERIMENT
```

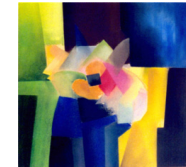
- Output:

```
Before a = 0.0
After a = 63.77
```

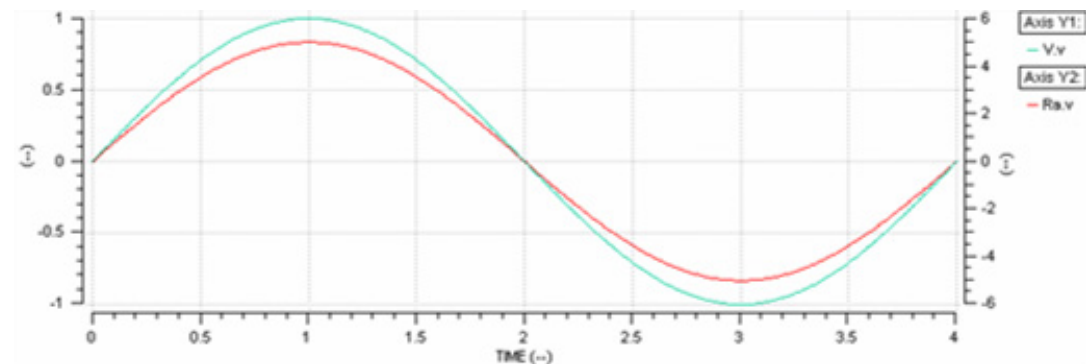
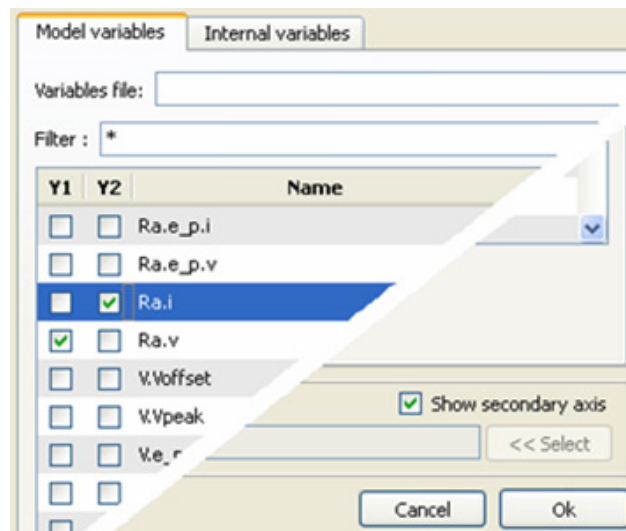


# Monitor Improvements

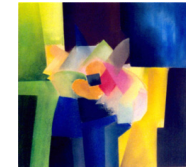
# New Secondary Axis for Plots



- **Variables displayed against two independent Y-axes**
  - It allows using two scales at the same time in order to plot variables with different order of magnitude
  - The user can display variables against different types of axis: logarithmic scale, decimal scale etc
- **Group of variables to be displayed on each axis are selected on contextual menu**




# New capability in Plots: Markers

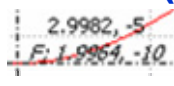


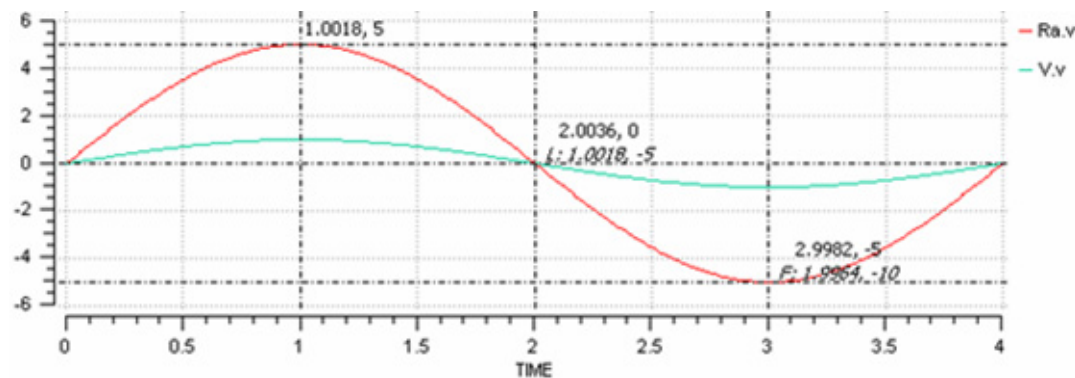
- A way to show a specific value of a point in a plot

- Types of markers:

- Simple marker: coordinates of a point in a plot 

- Comparison with last marker (L): comparison with the coordinates of the last added marker 

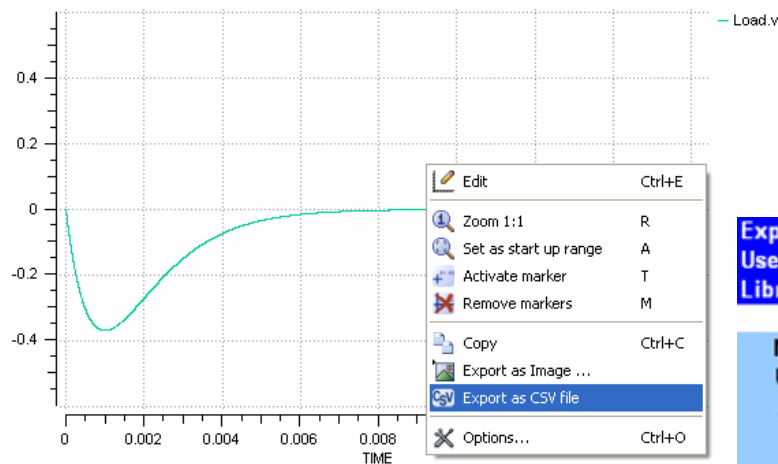
- Comparison with first marker (F): comparison with the coordinates of the first added marker 



# Exportation of CVS files



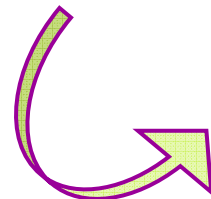
- Values of tables and each plot can be exported into CSV in order to transmit information between programs. Ex: from a database to a spreadsheet



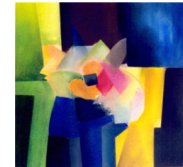
Exportation into Excel


Experiment: exp1 Library: ELECTRICAL\_EXAMPLES Component: Amplifier Partition: default  
 User: fct Date: 2010:06:16 Time: 11:07:08  
 Library dependency list: ALL V0.0

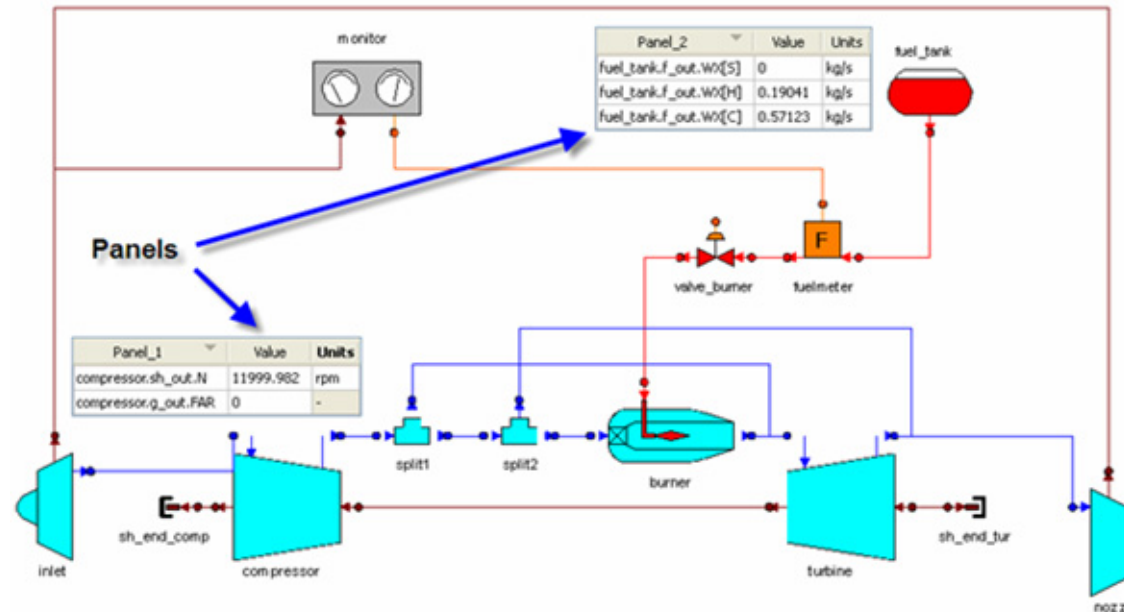
Name	#Calculation id	#ESI	#Status	RF.i	RI.R	RF.v
Units	--	--	--	--	Ohm	V
Alias	--	--	--	--	--	--
1	transient-1	0	INTEG_BEGIN	0	100	0
2	transient-1	0	IS_CINT	0.00015707	100	0.062829
3	transient-1	0	IS_CINT	0.00031411	100	0.125642
4	transient-1	0	IS_CINT	0.00047106	100	0.188425
5	transient-1	0	IS_CINT	0.0006279	100	0.251161
6	transient-1	0	IS_CINT	0.00078459	100	0.313835
7	transient-1	0	IS_CINT	0.00094108	100	0.376431
8	transient-1	0	IS_CINT	0.00109734	100	0.438935



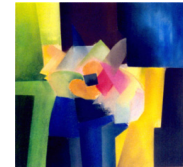
# New Graphical User Interface GSE



- **GSE is the name given to a new capacity software module included in the experiment monitor . It enables the user to interact with their simulation and help them to:**
  - Better their understanding of the simulation
  - Sort and classify the simulation results based on criteria such as the physical arrangement
  - Presentation of simulations go hand in hand with visual displays of the simulation background to be simulated



# New Graphical User Interface GSE



- **Elements in the GSE window:**

- **Background**

- The user can select any file with format .jpg, .bmp, .png...Controls are distributed in this delimited work area

- **Controls**

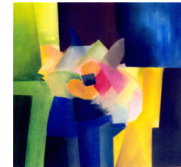
- **Panels:** model variables and their values are displayed there


Panel_2	Value	Units
fuel_tank.f_out.WX[S]	0	kg/s
fuel_tank.f_out.WX[H]	0.19041	kg/s
fuel_tank.f_out.WX[C]	0.57123	kg/s

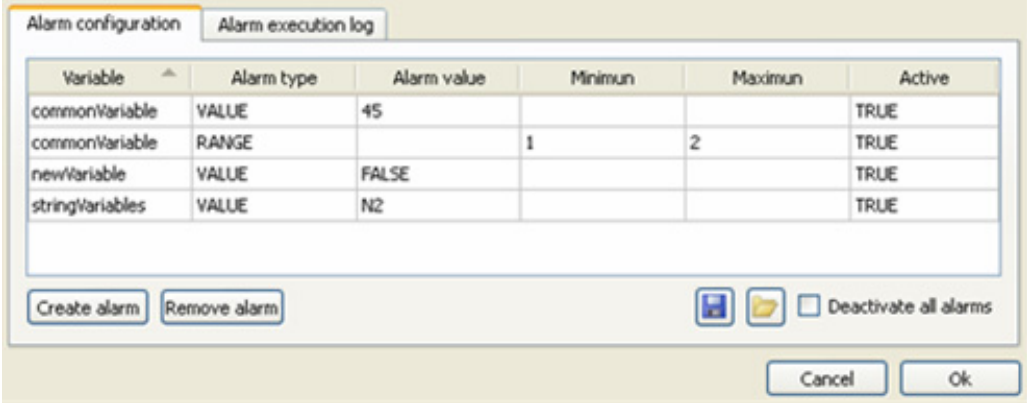
- **Navigators:** quickly access to related GSEs





# Alarms



- An alarm  is a mechanism that enables us to analyse the behaviour of a variable visually or in detail (by means of a log)
- Types of alarms
  - Value alarm: this alarm is triggered when a certain value is reached
  - Range alarm: it is triggered when the variable selected exceeds a numerical interval

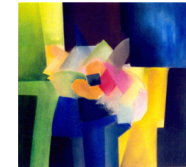


Variable	Alarm type	Alarm value	Minimum	Maximum	Active
commonVariable	VALUE	45			TRUE
commonVariable	RANGE		1	2	TRUE
newVariable	VALUE	FALSE			TRUE
stringVariables	VALUE	N2			TRUE

Create alarm Remove alarm    Deactivate all alarms

Cancel Ok

# Alarms



- All the alarm log activations and deactivations throughout the various model simulations are shown in the alarm execution log
- When we click 'Play' to start simulation, the alarms (as long as they are activated) start performing checks and whenever one of them is triggered it is graphically displayed in red on the Watch and in the GSEs

The screenshot displays the EcosimPro software interface during a simulation. On the left, the 'Alarms execution log' window shows a list of alarm events with their status, type, and range. The log starts with 'INIT execution of model:' and lists various 'Alarm ON' and 'Alarm OFF' events for variables like 'x[02]', 'x[04]', 'x[18]', 'x[03]', 'x[05]', 'b', 'col', and 'st' at different time intervals.

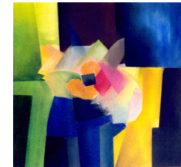
In the center, a process diagram shows a 'Filter' and a 'Pump1' connected by a pipe. The 'Filter' is represented by a diamond shape, and 'Pump1' is a circle. The diagram is overlaid with a grid.

On the right, the 'Watch' window shows a table of variables and their values. The table has columns for 'Index', 'Inputs', 'Value', 'Units', 'Category', and 'Type'. The 'Value' column shows values like 0, 0.97, and 0.97. The 'Category' column shows 'BOUNDARY', 'EPLUCOT', and 'EPLUCOT'. The 'Type' column shows 'REAL'. The table is titled '33.2'.



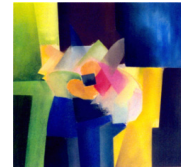
# Experiment Wizards

# Calculation Wizard

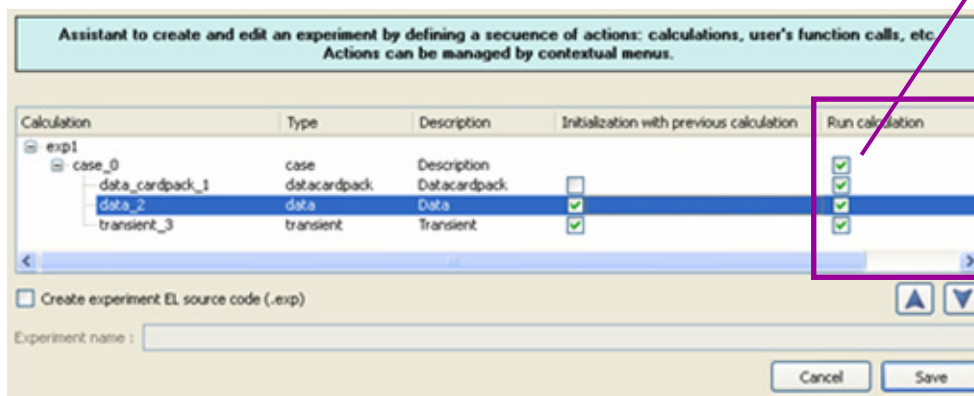


- **Calculations:**
  - Parametric
  - Steady
  - Transient
  - **Function Calls** *(NEW)*
    - The "Function Calls" option displays a window where it is possible to add and delete selected files which include the function calls and all the statements (REPORT,EXEC\_INIT, RESET, etc) we want to use at the moment of execution of the calculation defined in the existing calculation tree
  - **Data** *(NEW)*
    - The "Data" option displays a window with the model data where it is possible to assign values that will be introduced when the calculation defined in the existing calculation tree is executed
  - **Data Cardpack** *(NEW)*
    - The "Data CardPack" option displays a window with the model data where it is possible to assign values that will be introduced when the model is reset, after the call to RESET\_VARIABLES() (see (un)setCallAfterReset() function in EL Modelling Language Manual), and will therefore replace the values initialised in the model for any subsequent calculation defined in the existing calculation tree

# Calculation Wizard



- New option “Run calculation” to select which calculations of the experiment are run when the experiment is executed



If a case is not selected none of its calculations will be run

- Easily customization of experiments