

**THERMOPTIM®**

**USE**

**OF**

**EXTERNAL COMPONENTS**

**VERSION JAVA 1.5**

**© R. GICQUEL JULY 2003**

# SOMMAIRE

<b>USE OF EXTERNAL COMPONENTS.....</b>	<b>3</b>
EXTENSION SYSTEM FOR THERMOPTIM BY ADDING EXTERNAL CLASSES .....	3
SOFTWARE IMPLEMENTATION.....	4
<b>Viewing the available external classes.....</b>	<b>4</b>
<b>Representation of an external component in the diagram editor .....</b>	<b>5</b>
<b>Loading an external class .....</b>	<b>5</b>
<b>Thermocouplers .....</b>	<b>5</b>
EXTERNAL NODES.....	7
MISCELLANEOUS COMMENTS .....	8

© R. GICQUEL 1997 - 2003. All Rights Reserved. This document may not be reproduced in part or in whole without the author's express written consent, except for the licensee's personal use and solely in accordance with the contractual terms indicated in the software license agreement.

Information in this document is subject to change without notice and does not represent a commitment on the part of the author.

## Use of external components

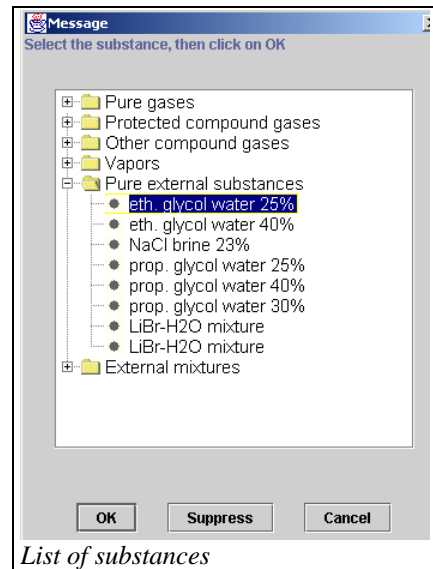
Volume 3 of the reference manual Thermoptim is dedicated to the achievement of external components. In this note, we just provide some guidance on their practical use.

### ***Extension system for Thermoptim by adding external classes***

One great advantage of Thermoptim is that its graphic environment can be used to visually build models for a large number of energy systems, from the most simple refrigerator to complex integrated gasification combined cycle electric power plants using several hundred elements.

Not only does this greatly simplify the modeling process and facilitate subsequent use and maintenance of the model, but it also makes the models more reliable. The connections between the different elements are made automatically, thus ensuring consistency.

Through version 1.4, only the components available in the Thermoptim primitive type set could be assembled in this manner, which limited the potential of the software. A number of users wished to be able to define their own elements and/or their own substances.



Thermoptim's interface with external classes (Java code elements) provides the solution and facilitates the interoperability of the software with the outside world, especially with other applications developed using Java.

The benefits are two-fold:

- create Thermoptim extensions from common primitive type set, by adding external modules that define the elements that automatically appear on the screens in a seamless fashion. Thus users can add their own substances or components not available in the basic set.

- drive Thermoptim from another application, either to guide a user (smart tutorial) or to check the code (driver or regulation, access to thermodynamic libraries).

The previous and opposite figures show how the external substances are added to Thermoptim's list of substances, and then replace an internal substance on the point screen. They are just as easy to use as if they were part of the software.

*Substance screen*

The figure opposite shows how an external component representing a solar collector appears in the diagram editor, and the figure below shows the screen of the corresponding process, composed partly of ThermoOptim internal code and partly of external code (lower third on the right).

### Software implementation

Practically speaking, adding a new external process is quite easy. Simply create a specific class, which inherits from the abstract parent class `extThopt.ExtProcess`. The interaction with ThermoOptim is ensured on two levels:

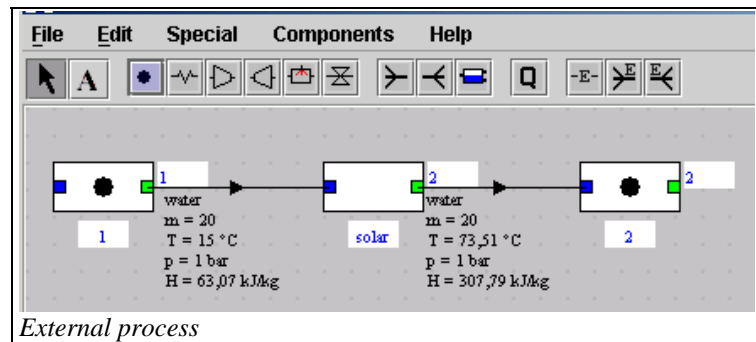
- by general methods for performing the required calculations;
- by a JPanel that is built in to the external process screen. Thus, the class designer can create his own graphic interface, which is then inserted into the ThermoOptim screen.

The classes used in ThermoOptim, `CorpsExterne`, `TransfExterne` and the other, define the interface methods, which fall into three additional categories:

- Internal ThermoOptim methods designed to be obfuscated
- Non-obfuscated internal methods that are directly used by the external classes: These methods are used to execute the ThermoOptim methods with external parameter settings, and **must not be overloaded**. Their signature must be strictly respected: in particular, the structure of the Vectors of Objects used as arguments must correspond exactly to what ThermoOptim expects, or a `ClassCastException` will be generated.
- External methods corresponding to ThermoOptim's obfuscated methods: These methods are used to execute the external methods from within ThermoOptim, and **must be overloaded**.

The non-obfuscated methods falling into the last two categories are listed in Appendix 1, for all of the accessible classes. For more information, please see the html documentation for these classes provided in the folder `api_ThermoOptim`.

### Viewing the available external classes






To help you use and manage the external classes, the line External Class Viewer from the Special menu displays all of the external classes available.

They are sorted by type (substances, processes, mixers, dividers, drivers) with a short description of the class selected and where it comes from (extThopt.zip and extUser.zip archives as well as classes under development).

This screen can be consulted while you are developing your model.

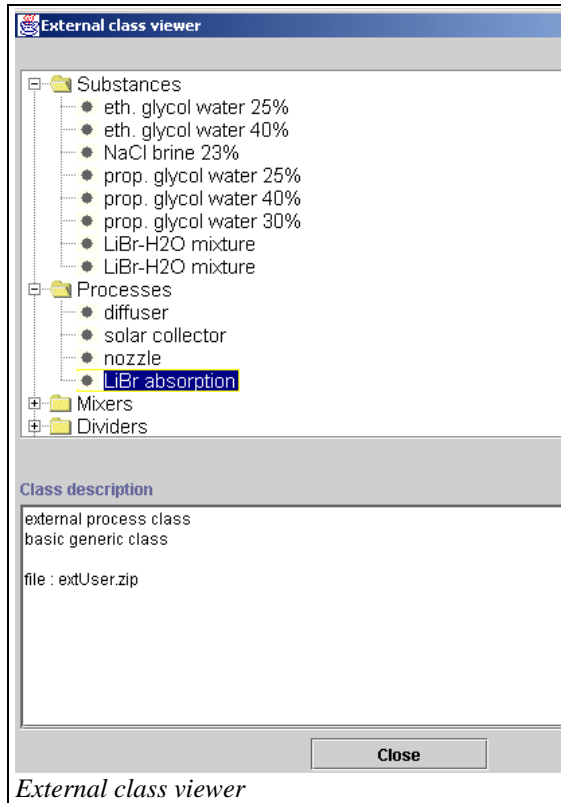
## Representation of an external component in the diagram editor

Specific icons were added to represent the external components ( for processes,  for mixers, and  for dividers). The external component is then selected when the simulator is updated from the diagram as indicated below.

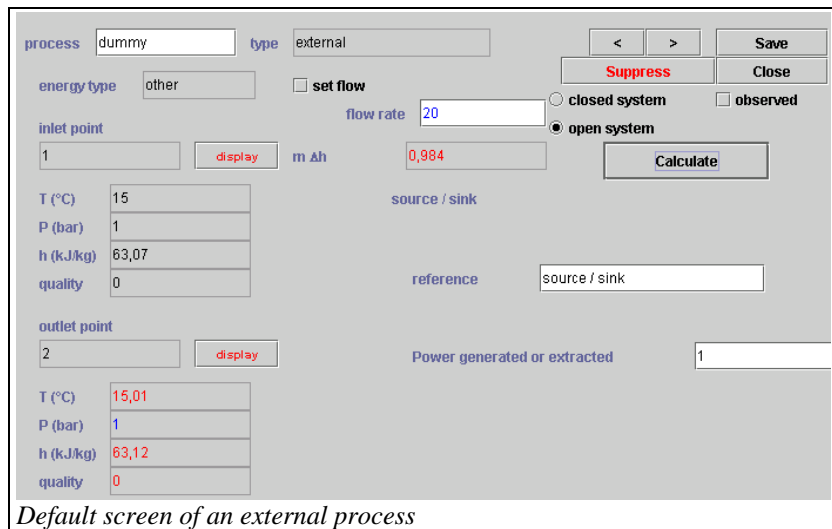
## Loading an external class

To load an external process (for an external node, the process is the same), you can either:

- from the simulator screen click on the column header of the process array, then choose External and finally select the type of external process you want from the list;
- or, from the diagram editor, build the external component graphically then update the simulator from the diagram. In the case of an external process, by default it is a “heat source / sink” type, as shown in the screen opposite.



External class viewer



Default screen of an external process

Once this default process is created, double click on the label “source / sink” to access the list of all external processes available. Choose the one you want and it will be loaded.

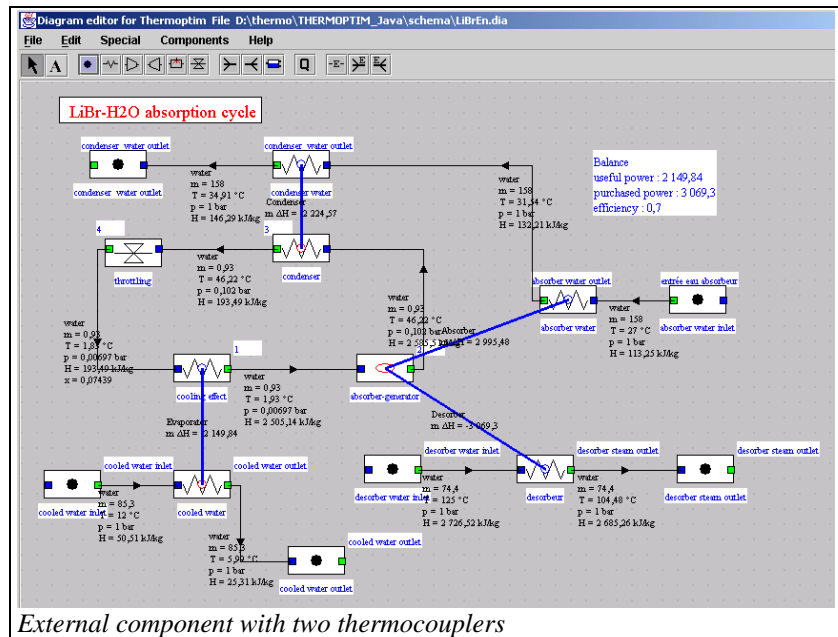
## Thermocouplers

The thermocoupler system completes the heat exchanger system by allowing components other than exchange processes to connect to one or more exchange processes to represent a thermal coupling. This system does not encompass the exchanger system: two exchange processes cannot be connected by a thermocoupler.

This system has a number of benefits, because it can be used to represent many thermocouplers that do not constitute a heat exchange in the traditional sense, like for example cooling the walls of the combustion chamber

of a reciprocating engine, cooled compression, and above all supply or removal of heat from the multi-functional external components.

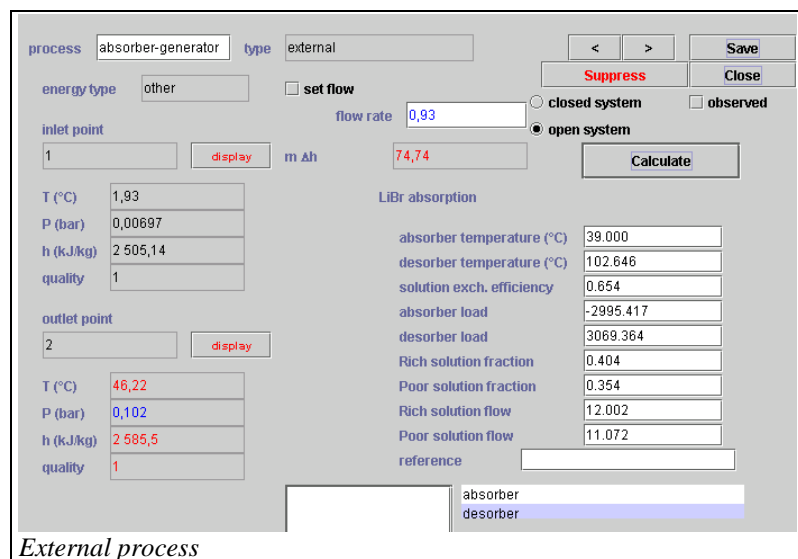
The figure above is an illustration of this: An absorption refrigeration cycle, whose absorption-desorption system is defined and integrated in an external process, is supplied with the steam that exits the evaporator then enters the condenser. This cycle involves the mixture LiBr-H<sub>2</sub>O, whose properties are modeled either directly in the external process, or in an external substance, and requires high temperature heat supply to the desorber and medium temperature heat removal from the absorber. The representation of these thermocouplers is possible thanks to the thermocoupler system: the external process calculates the thermal energies that must be exchanged, and the thermocoupler recalculates the corresponding “exchange” process, which updates its downstream point.



External component with two thermocouplers

The types of thermocouplers used by an external component appear in the lower right hand corner of the screen. Double click on one of the types to open the screen of the corresponding thermocoupler.

Given that thermocouplers are a type of heat exchanger, it is valuable to define them by values such as effectiveness  $\epsilon$ , UA, NTU or LMTD, that can be calculated using similar equations. The component sends to each of its thermocouplers the equivalent values for flow rates, inlet and outlet temperature and thermal energy transferred, which they must take into account in their calculations. Specific methods are provided in the external class code and are not user-modifiable.



External process

However, there are limits to the similarities with exchangers: for example, temperature crossovers unacceptable in an exchanger may occur in a thermocoupler, leading to absurd values.

So it is best to transmit values that are unlikely to lead to this type of situation. One possible solution is to assume that the thermocoupler is isothermal for calculations of characteristics that are similar to exchanger characteristics. For example, a combustion chamber may be assumed to be at mean temperature between its upstream and downstream points when calculating cooling. This assumption is not absurd and may prevent a temperature crossover between the cooling fluid and the gases that cross the component.

In the case of the absorption machine presented above, we assumed that the absorber and desorber were isothermal.

Both lead to the screens below. If we had not taken the temperature of the absorber as a reference for the exchange calculations, keeping the temperatures of the steam entering and exiting the external process, we would have ended up with a temperature crossover.

For external processes that accept several thermocouples and for external nodes, the potential complexity of the calculations prevents the exchange process from driving the thermocoupler. Its load is always set by the external component. This is why there are fewer options for calculating a thermocoupler than for a heat exchanger: The user can only choose between calculating the outlet temperature of the exchange process (at a given flow rate) and the flow rate, when the temperature is known.

**Absorber**

Parameter	Value
name	Absorber
type	counterflow
thermal fluid	eau absorbeur
process	absorbeur-générateur
Te	27
Ts	31,53731177
m	158
Cp	4,17841059
m ΔH	2 995,48274326
Te (calculated)	39
Ts (calculated)	38,9
m (calculated)	12,00144475
Cp (calculated)	2 495,88090821
m ΔH (calculated)	-2 995,48274326
UA	318,41186499
R	0,0220399605
NTU	0,482304198
LMTD	9,50942118
epsilon	0,381286703

Note that on the thermocoupler screen, the external component fluid can be selected as a pinch fluid and a minimum pinch value can be entered (see optimization method, volume 1).

### External nodes

The nodes of Thermoptim's basic set are extremely simple components used to complete the description of the fluid circuits. They are always adiabatic, and they ensure the conservation of the mass flow rate and enthalpy

**Desorber**

Parameter	Value
name	Desorber
type	counterflow
thermal fluid	désorbeur
process	absorbeur-générateur
Te	125
Ts	104,4765547
m	74,4
Cp	2,01009617
m ΔH	-3 069,30494785
Te (calculated)	102,846
Ts (calculated)	102,746
m (calculated)	11,07144475
Cp (calculated)	2 772,32386757
m ΔH (calculated)	3 069,30494785
UA	375,39331172
R	0,00487238402
NTU	2,51013315
LMTD	8,17623722
epsilon	0,918110642

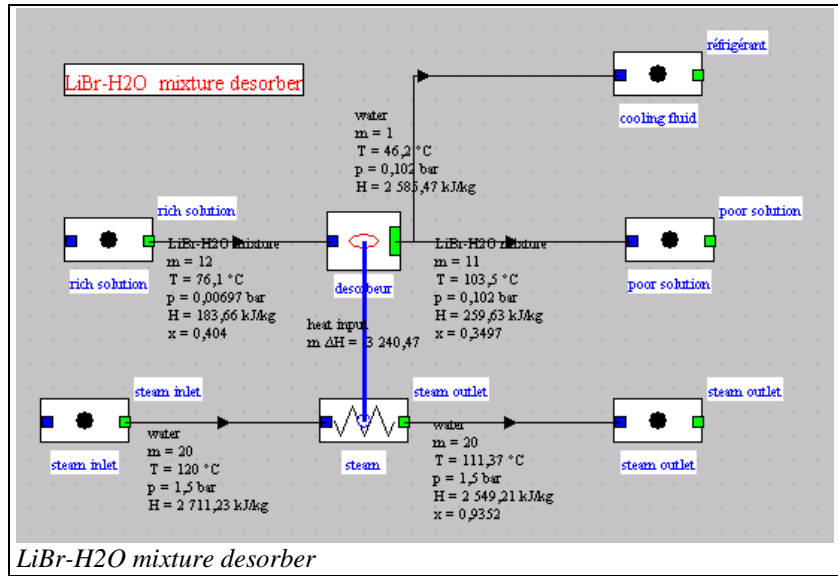
However, there is one somewhat special component, considered as a process, but which is in fact a node in most cases: combustion, which receives an oxidizer and a fuel, and from which burned gases exit.

In the LiBr-H<sub>2</sub>O absorption example, a number of energy systems are involved, as well as components of varying complexity, which can have a number of input and outlet fluids, after various internal calculations, with or without thermocoupling with external heat sources.

External nodes allow a user to define these components. Only external mixers and dividers are defined: no component simultaneously performs both functions (receiving and emitting several fluids), but you need only couple an external mixer with an external diffuser to do so.

In many respects, we encounter the same problems in implementing these external nodes as with thermocouplers: the potential complexity of the calculations to be made in the node makes it necessary for the node to take over and control both the main vein and the branches, since no default calculation is possible.

The verification and consistency problems are even more critical than for thermocouplers: only the node designer knows which processes it must be connected to in inlet and outlet. The user must refer to the documentation of the class to know how to use it.



The figure above shows the diagram of a desorber for an absorption machine using the LiBr-H<sub>2</sub>O mixture whose properties must in this case be provided by an external substance.

The external node screen is shown opposite. As with external processes, it contains a general part defined in the Thermoptim basic set, completed by a part defined by the user (here the lower left zone). In the model shown here, the only parameter defined in the node is the temperature of the desorber. The properties of the rich solution (mass fraction and flow rate) and the state of the refrigerant are obtained from their respective processes.

The node calculates the flow rates of the refrigerant and the poor solution, its state, and heat supply required.

The screenshot shows the 'desorbeur' node configuration in Thermoptim. The node type is 'external divider'. The main process is 'desorbeur' with a display button. The rich solution is 'rich solution' and the iso-pressure checkbox is unchecked. The global parameters are  $m_{global} = 12$ ,  $h_{global} = 183,65688559$ , and  $T_{global} = 76,1$ . A table displays the process data:

process name	m abs	m rel	T (°C)	H
cooling fluid	1,0013	1,0013	46,2	2 585,47
poor solution	10,9987	10,9987	103,5	259,63

Additional parameters include 'desorber temperature (°C)' set to 103,500, 'Rich solution fraction' set to 0,404, 'desorber load' set to 3240,469, and 'Poor solution fraction' set to 0,350. Buttons for 'Duplicate', 'Suppress', 'Save', 'Close', 'Calculate', 'add a branch', and 'delete a branch' are also visible.

*LiBr-H<sub>2</sub>O mixture desorber screen*

The thermocoupler then calculates the final state of the steam used.

Before each recalculation, the node checks that its structure is correct and loads the values that it needs to be calculated. Then it updates the thermocoupler that can in turn be recalculated. In this example, we assumed that the desorber was isothermal, and we took the flow rate of the poor solution as the reference flow rate.

### Miscellaneous comments

You may have noticed that the screens of the external processes represented here are composed partly of Thermoptim internal code and partly of external code (lower right section in the processes, lower left in the nodes.). This is because the character strings used in the external components have not been "internationalized" the way the Thermoptim basic set has.



