Heating Systems

User Guide

DISTECH CONTROLS™

an AcuityBrands company

Innovative Solutions for Greener Buildings™

Legal

TABLE OF CONTENTS

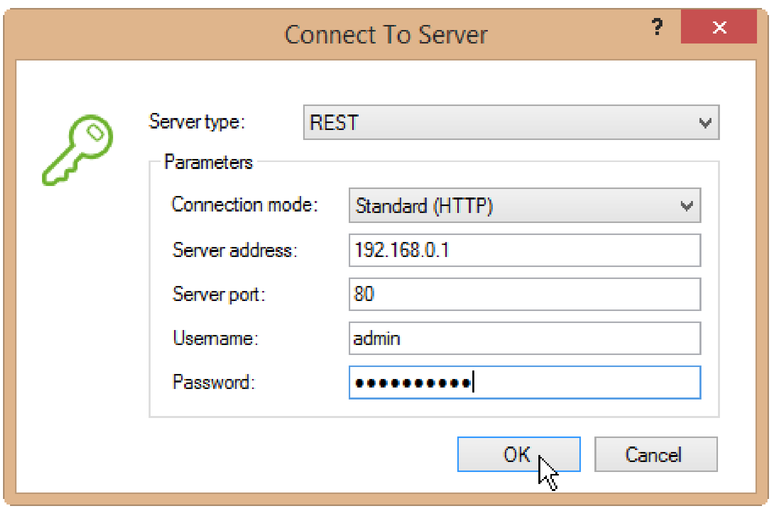
Accessing xpressENVYSION Graphical Templates

Prerendered graphical templates are provided to monitor and configure parameters specific to the heating application being used. The graphical templates are compatible with xpressENVYSION and can be accessed remotely through a web browser. The procedure to load an EC-*gfx*Program code and its corresponding xpressENVYSION template is outlined below.

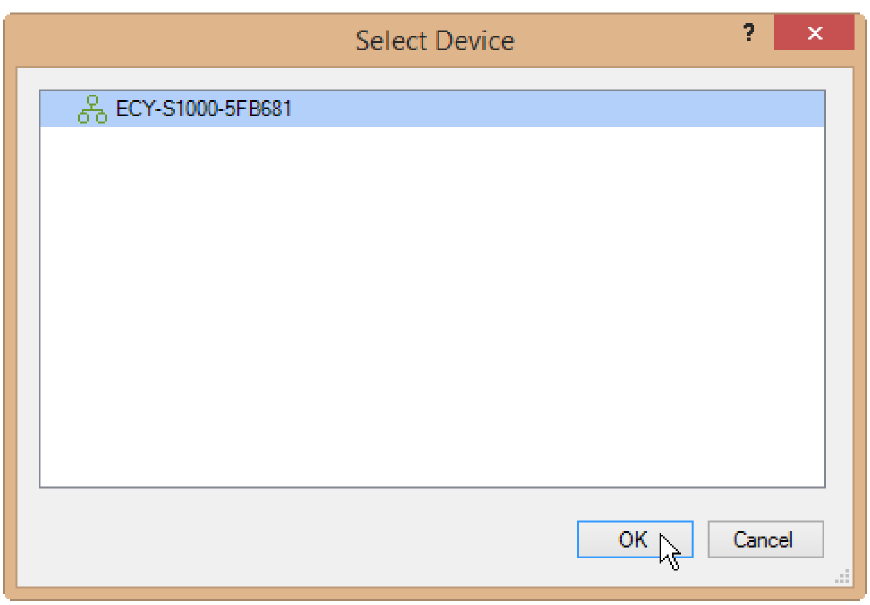
Loading an EC-gfxProgram code

Before the xpressENVYSION templates can be used, the controller must be loaded with the appropriate EC-*gfx*Program code. This can be done using either EC-*gfx*Program, or xpressNetwork Utility (please see the *xpressNetwork Utility User Guide* for more information).

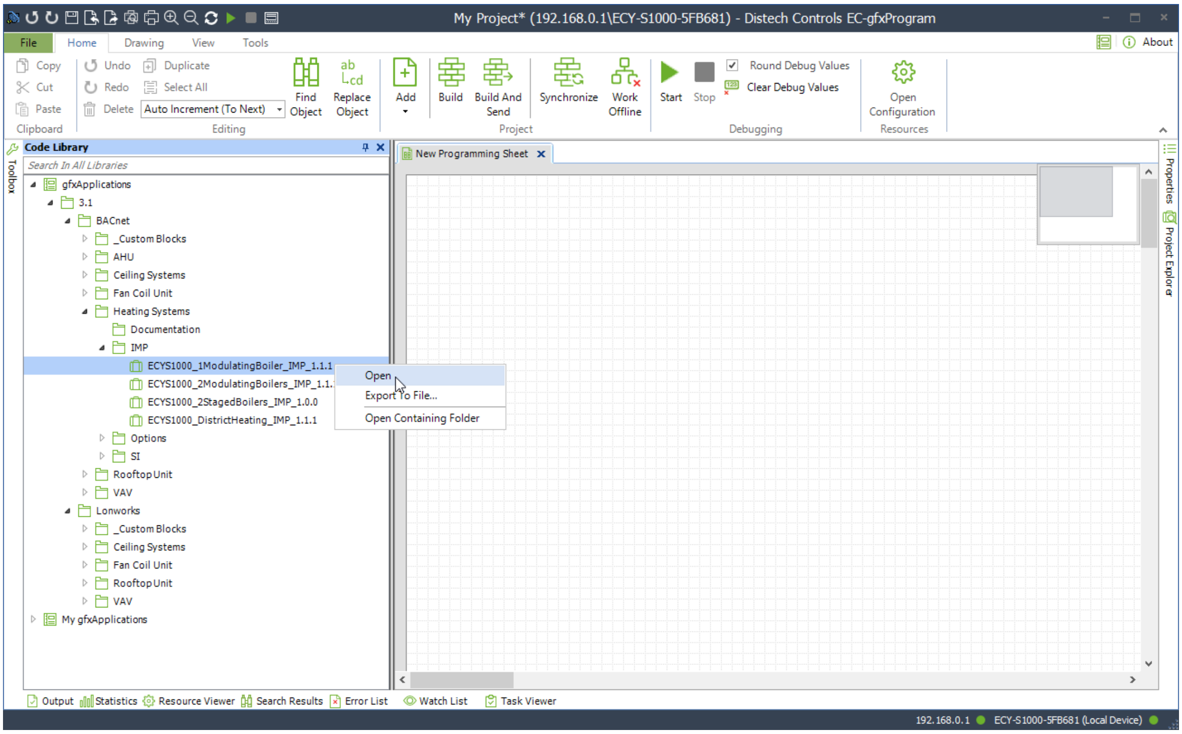
1. Start EC-*gfx*Program. Under the **File** tab, click **Connect to Server** and connect to the required controller using the correct user name and password.



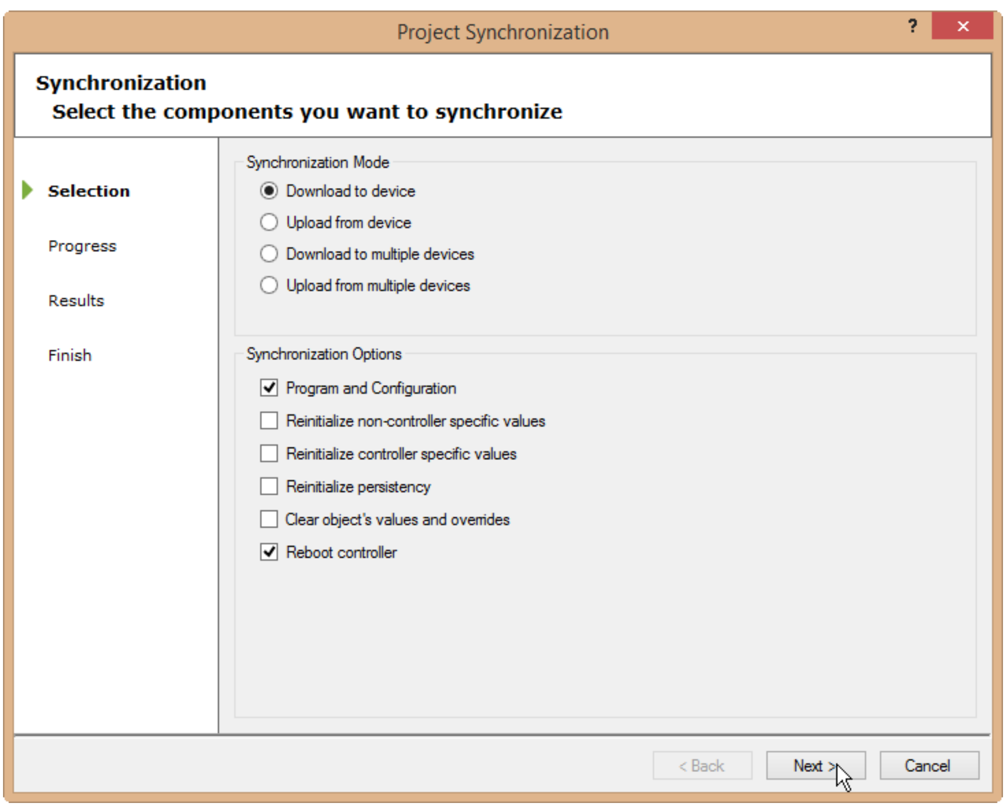
2. Select your device and click **OK**.



3. In the Code Librabry pane to the left, navigate to the Heating Systems folder and right click on the desired project and click **open.** Alternatively, you can drag and drop the desired project onto the programming sheet. In this example, we will choose 1 Modulating Boiler with imperial units.



4. Once the code is added to the programming sheet, it must be synchronised with the controller. Click the **Synchronize** button located on the top ribbon. Choose **Download to Device**, **Program and Configuration**, and **Reboot Controller** before clicking next to start synchronization.

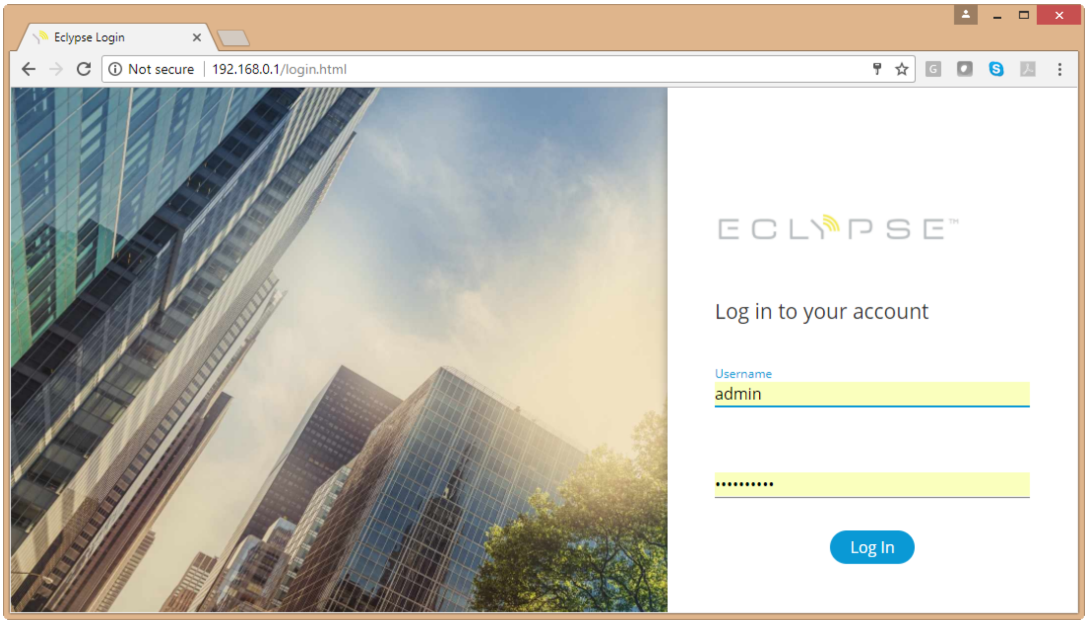


Once the Synchronization is done, you must wait for the controller to come back online. If you connect wirelessly, you must reconnect to the controller via wifi.

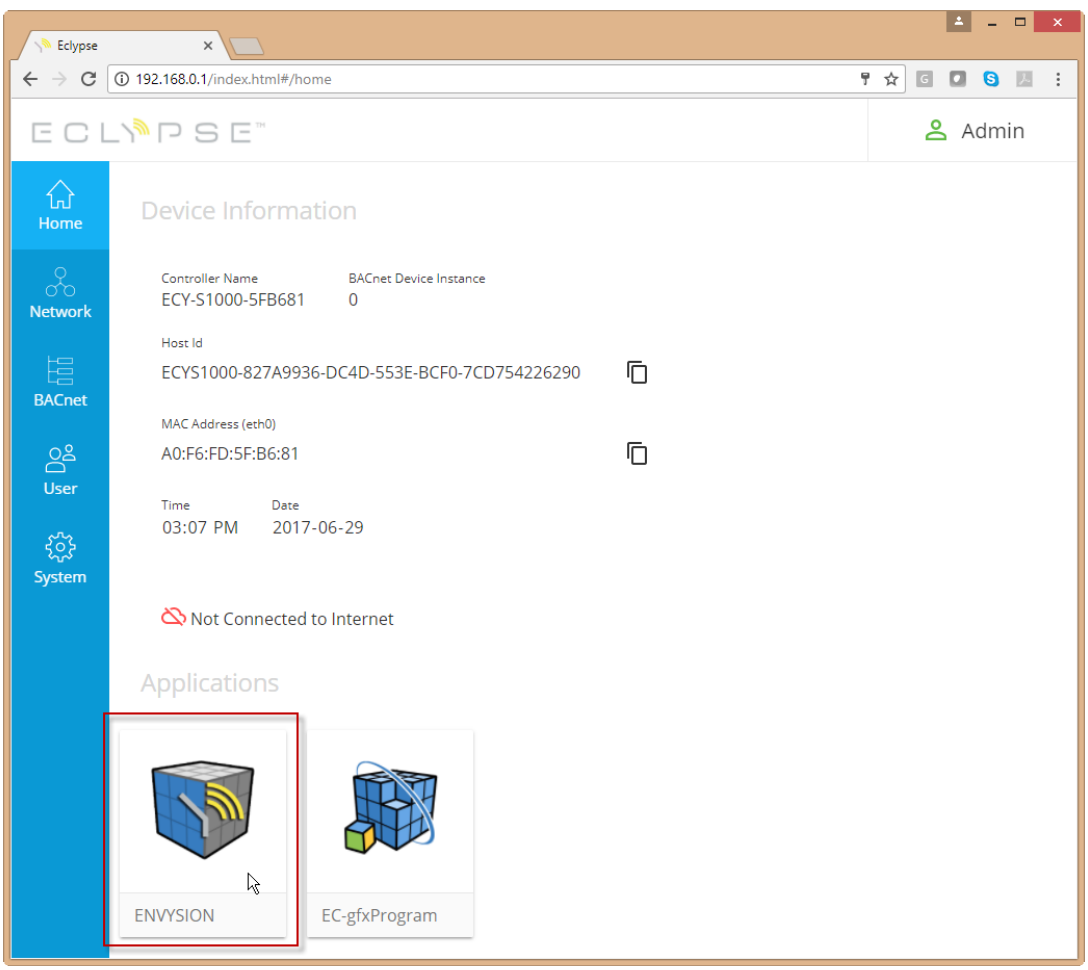
Loading an xpressENVYSION Template

Once the EC-*gfx*Program code has been successfully loaded and synchronized with the controller, we can now load an xpressENVYSION graphical template.

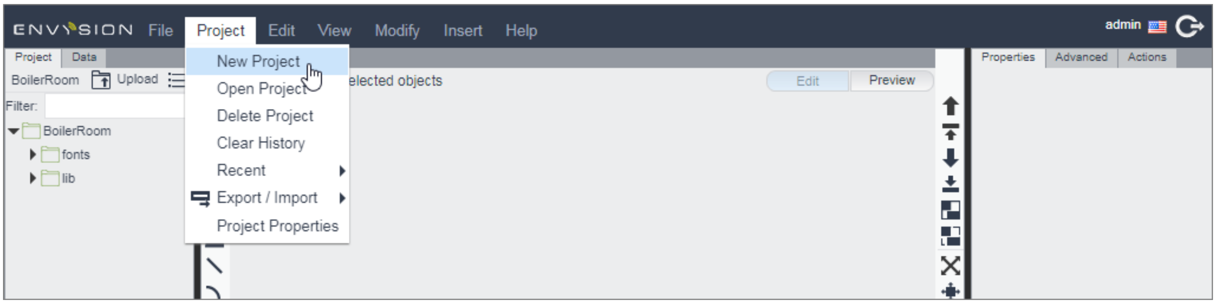
1. Within your web browser, navigate to the controllers login page and input your credentials.



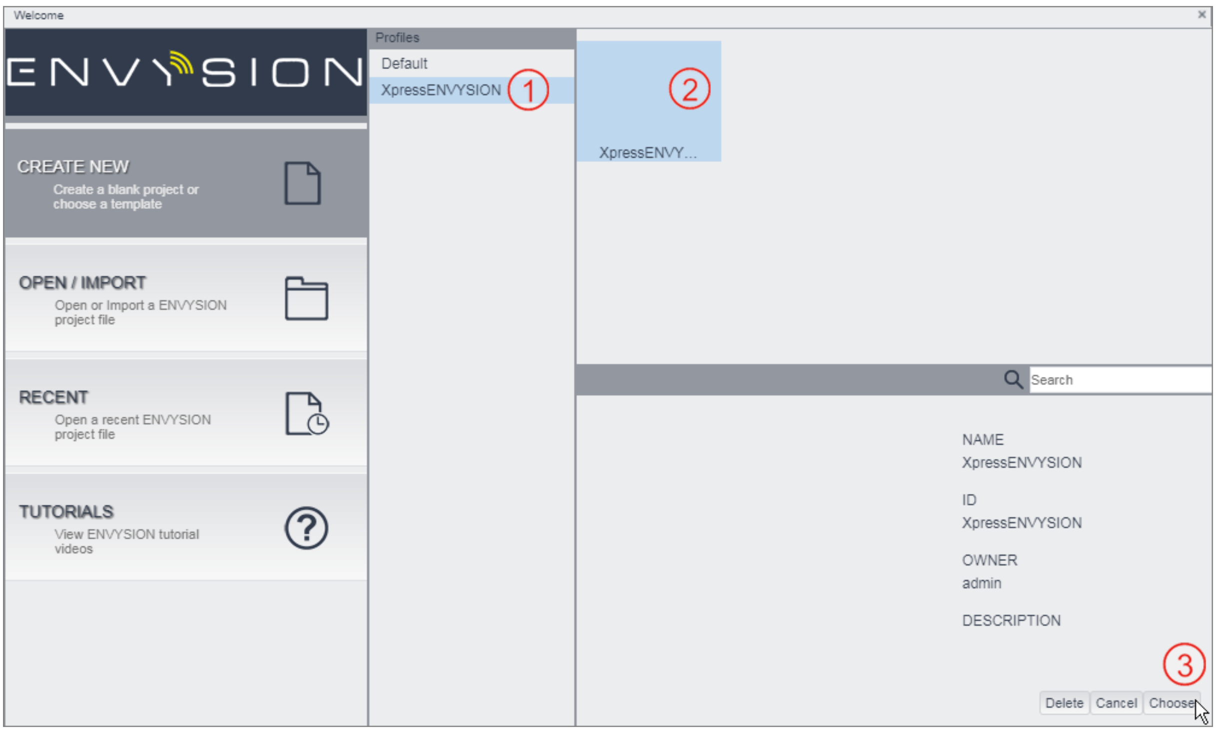
2. In the **Home** tab, click the **ENVYSION** tile. ENVYSION will start in a new browser tab.



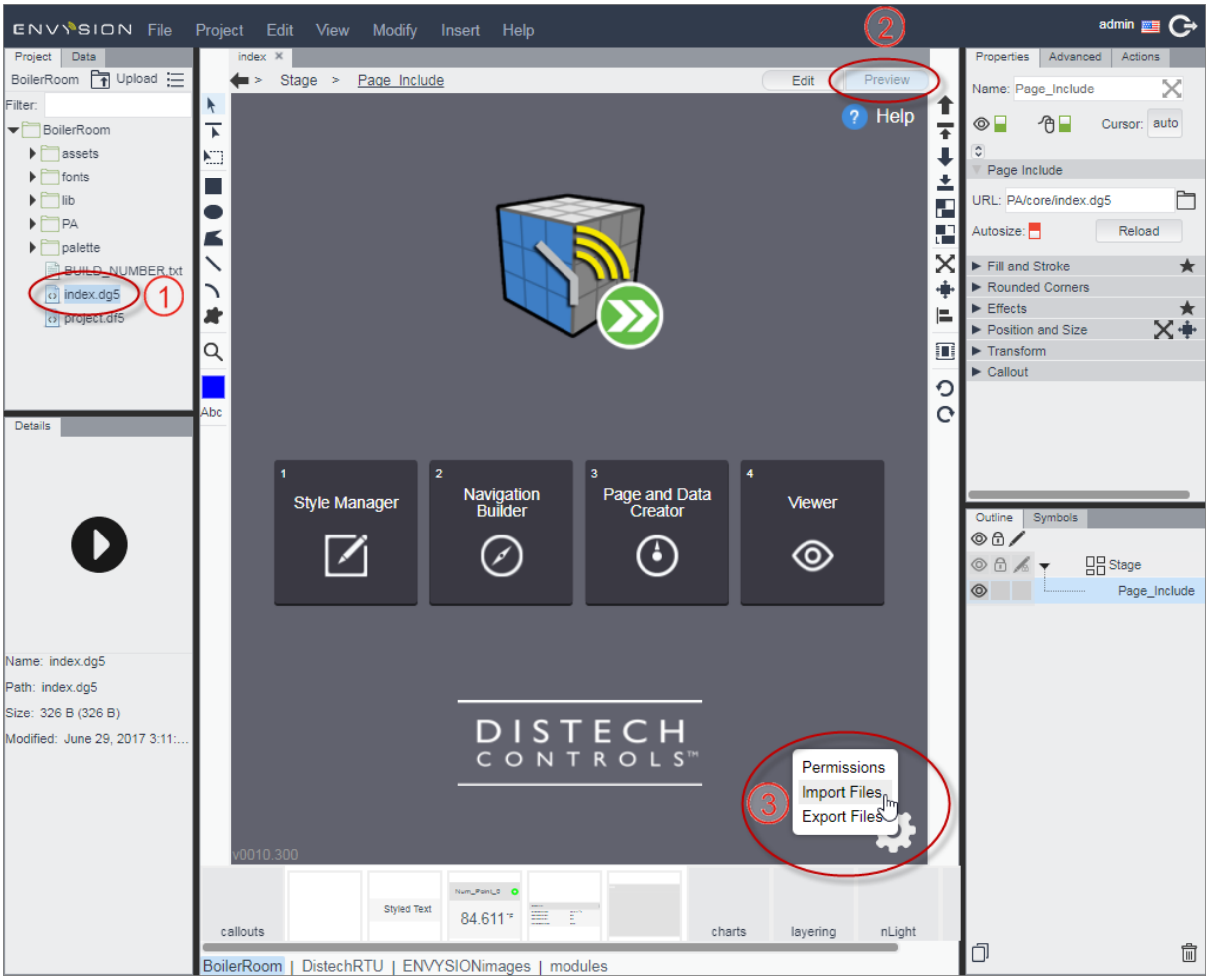
3. From the **Project** tab choose **New Project**.



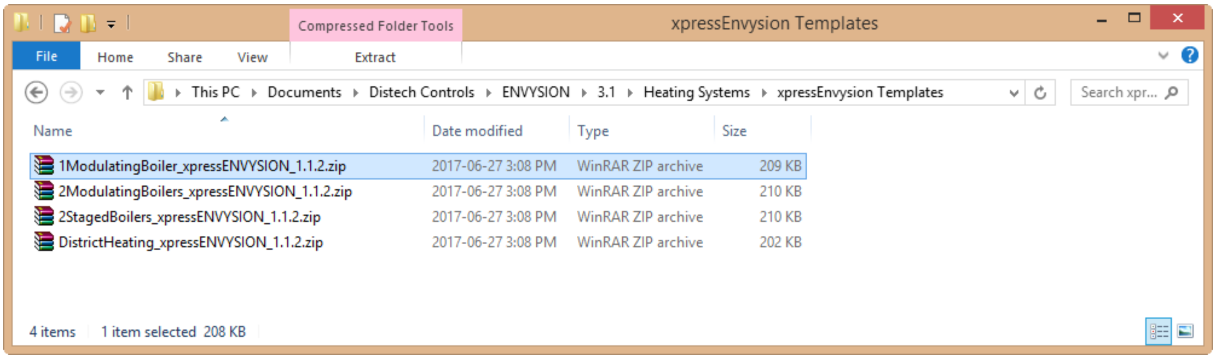
4. Under the **Profiles** column choose xpressENVYSION (1), choose the xpressENVYSION project to the right (2), finally click **choose** (3). Input a name for your project and click **Ok**. The xpressENVYSION project will load.



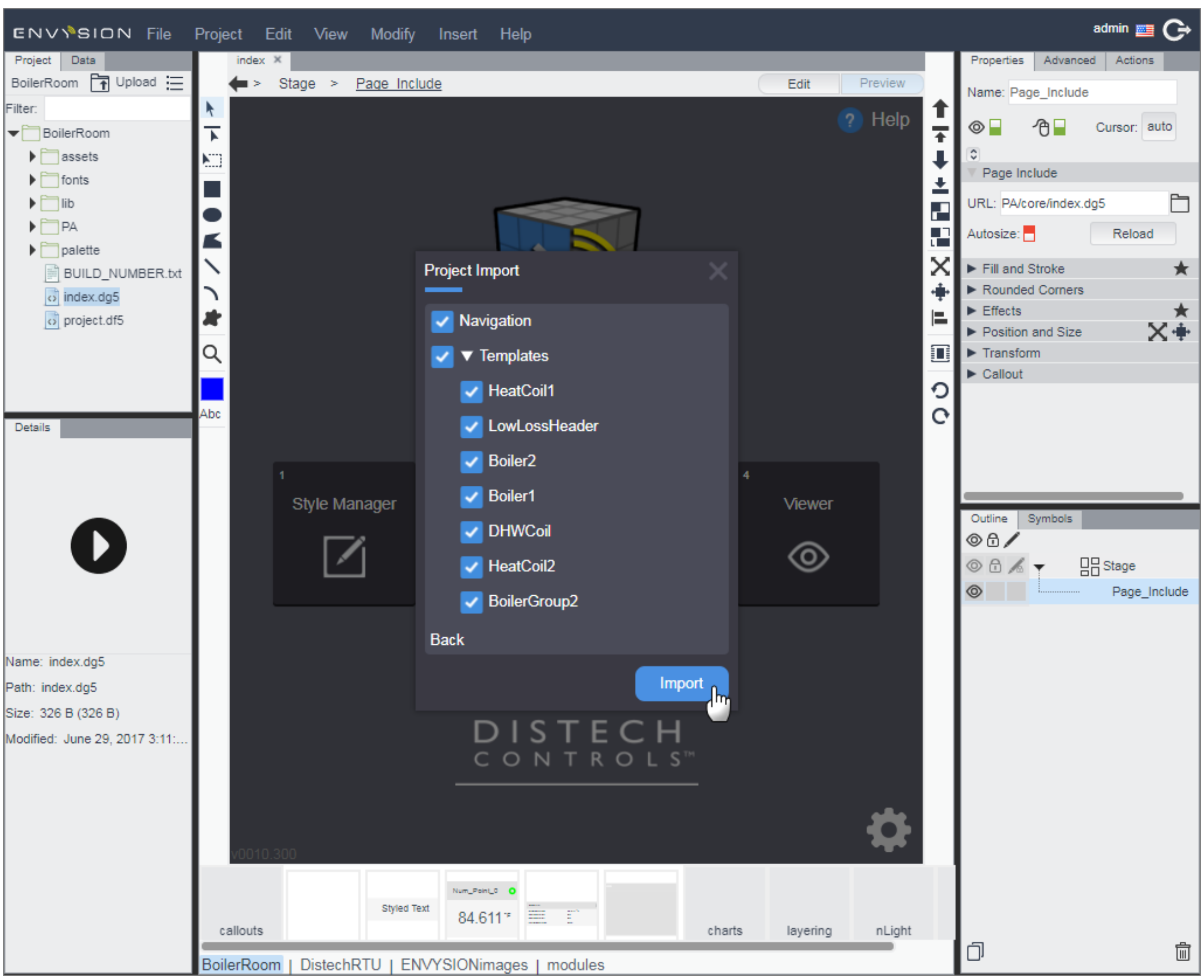
5. Once the new xpressENVYSION project has been loaded, we must now import the corresponding template. From the project pane on the left, double click the **index.dg5** file (1), next click **preview** (2), lastly click the gear icon on the bottom right and choose **Import Files** (3). The project import window will appear.



6. Here we must choose the corresponding xpressENVYSION template. After installing the Productivity Enhancing Tools, the system will automatically place the required files in the user's documents folder:
   (\Documents\Distech Controls\ENVYSION\3.1\Heating Systems\xpressEnvysion Templates).

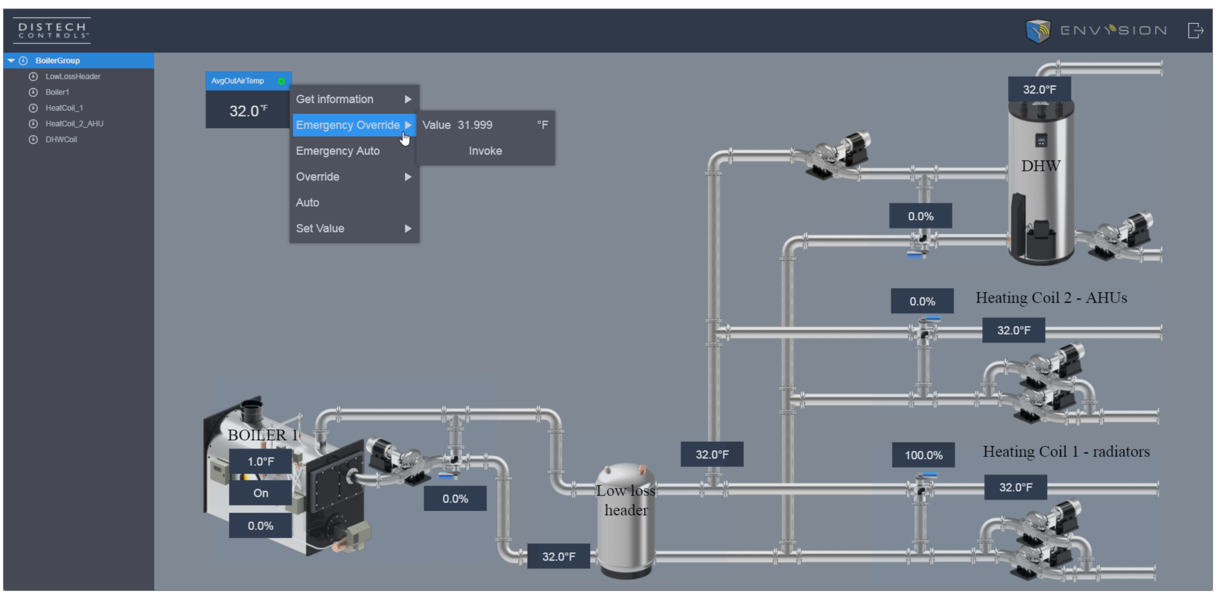


For our example we must choose the **1 Modulating Boiler** template which corresponds with the EC-*gfx*Program code we installed previously. Drag and drop the **1ModulatingBoiler_xpressENVYSION_1.1.2.zip** file onto the Project Import window. Click **Import**. The xpressENVYSION template will load in the browser.



7. Choose the **Viewer** tile.

8. Once the viewer is open, click the **BoilerGroup** in the panel to the left to see the graphics. From this view, parameters can be adjusted using the Point Status icon .



Heat Generation

In heat generation systems two major solutions can be distinguished, Boiler Systems and District Heating Systems.

The signal types used can be both logically or physically connected to the controller.

Signal Type	Description
AI/AO	Analogue input/output
DI/DO	Digital input/output
MI/MO	Multistate input/output
BI/BO	Binary input/output
Sch	Scheduler signal
Cal	Calendar signal

Standalone Boiler Control Block (HTG_BoilerStandAlone)

Boilers are the most common heat source, providing heating for standalone buildings which are not connected to the district heat distribution installation.

Boilers can be divided into groups based on different functional features.

Type	Feature	Description
Boiler Type	Standard	Requires return water temperature high enough to avoid vapour condensation.
	Condensation	Requires low return water temperature in order to achieve condensation and boost efficiency.
Fuel Type	Gas	Usually from a supervisory controller perspective, gas and oil boilers are operated in a similar way. Its built-in burner controller provides a basic sequence of burner operations.
	Oil	
	Solid Fuel (generally not used in the automated boiler rooms)	
Burner Type	Stage burners	Usually 1 or 2 stages.
	Modulated burners	
Size (water capacity)	Large Boilers	Can be started with valves closed and pump not running, and starts circulation after water temperature reaches a minimum value.
	Small Boilers	Cannot be started before water circulation is confirmed.

Usually boilers and burners are equipped with dedicated control systems which provide basic controls and all the necessary security functions. Boilers that have the option of external control must be supplied with a build-in controller which is able to accept commands from external control systems that can send alarms and confirmations. This approach usually relieves the automation system programmer from the obligation of providing low level burner sequence programming.

The standalone boiler control block can be configured to operate with staged on/off (1 or 2 stages) or modulated boilers.

Inputs

The Standalone Boiler Control Block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Boiler Water Temperature Sensor	Temperature value of the water at the boiler output (AI).
Heat Demand	Supply water temperature request from the heat consumer blocks (AI).
Boiler Alarm	Burner or boiler internal fault signal, active state results in immediate boiler switch off (BI).
High Temperature Thermostat	One or two stage thermostat that is used to stop boiler operation in case the temperature inside of the boiler rises too high. If the thermostat has two stages, the first stage is auto reset and the second stage requires manual reset directly on the boiler (BI).
C.BoilerPowerStage1/2	Mandatory configuration parameters, defining burner power levels for stage 1 and 2. (AI).
Reset	Alarm reset command (BI).

Optional Inputs

Input Parameter	Description
External Boiler Command Stage1/2	If connected, it disables internal boiler temperature control and transfers external commands directly to the stages command section of the code. However, all the security and time based sequences remain operational. It is used if Standalone Boiler Control block is used as a subordinate for group control block (BI).
External Boiler Modulated Control	If connected, it disables internal boiler temperature control and transfers external modulation command directly to the stages command section of the code. However, all the security and time based sequences remain operational. It is used if Standalone Boiler Control block is used as a subordinate for group control block (AI).
Boiler Return Water Temperature Sensor	Temperature value of the water entering the boiler (AI).
Boiler Pump Status	Confirms operation of the pump (feedback from a contact or a flow switch). Do not use simultaneously with Pump Start Command Feedback (BI).
Boiler Pump Command Feedback	Confirmation signal from the output of the controller that it is energized. Do not use simultaneously with Pump Start Command Feedback (BI).
Boiler Pump Fault	Fault signal of the pump (feedback from a contact or a flow switch) (BI).
Low Water Level Sensor	Fault signal used to block boiler operation if there is no water in the system (BI).
Isolation Valve Switch	Feedback signal from 2-way isolation valve used to confirm valve opening (BI).
Pump Exercise	Activation command for periodic pump and valve motorisation (BI).
Boiler Enable	Boiler start signal. Might be omitted if *ExtBoilerCmdStage1* is used (BI).

Outputs

The Standalone Boiler Control Block output signals (hardware and logic).

Output Parameter	Description
Boiler Command Stage 1/2	Boiler stage enable output (BO).
Boiler Modulation Control	Boiler modulation control output (AO).
Boiler Valve Control	Boiler control/isolation valve output (AO).
Boiler Pump Command	Start signal for primary side pump (BO).
Boiler Pump Status	Multistate pump status point (MO).
Boiler Pump Run Alarm	Fail to stop alarm active (BO).
Boiler Pump Stop Alarm	Fail to start alarm active (BO).
Boiler Pump Fault Alarm	Direct fault alarm active (BO).
Boiler Alarm	Direct burner/boiler fault alarm active (BO).
High Temperature Alarm	*HighTempThermostat* trip alarm active (BO).
Low Water Level Alarm	Low water level alarm active (BO).
Valve Closed Alarm	Isolation valve failed to open alarm active (BO).
Boiler Overheat	Boiler overheat priority signal (BI).
Stage Overheat Active	Boiler stages are switched off due to too high value of the *BoilerWaterTemp.* It is used if Standalone Boiler Control block is used as a subordinate for group control block (BO).
Stage On/Off Delay Active	Boiler stages change is locked by internal on/off delay. It is used if Standalone Boiler Control block is used as a subordinate for group control block (BO).
Effective Heat Demand	Effective heat demand value (AO).
Effective Secondary Water Temp Set Point	Effective supply water temperature set point value (AO).
Effective Return Water Temp Set Point	Effective boiler return-water temperature set point value (AO).
Active Stage	Information internal operation stage of the boiler (AO).

Block Functions

Boiler Startup and Stopping

When this block receives the *BoilerEnable* command (optionally if *ExtBoilerCmdStg1*is activated), it activates the startup procedure if there are no alarm conditions active.

For 2-way valve (or no valve) (*C.3WayValve* = FALSE)

1) Starting large water capacity boiler (*C.LargeVolBoiler*= TRUE):

- Burner is started.
- When boiler temperature reaches defined temperature, valve is opened.
- When valve is opened pump is started.

2) Starting small water capacity boiler (*C.LargeVolBoiler*= FALSE):

- Valve is opened.
- When valve is opened (confirmed by valve end switch) pump is started. If end switch is not available, a delay time of 120s is applied.
- When water circulation is confirmed, the burner is started

3) Stopping procedure (*BoilerEnableCommand* = off)

- Burner is stopped.
- After a pre-defined delay, the pump is stopped and valve is closed.

For 3-way valve (*C.3WayValve* = TRUE)

1) Starting large or small water capacity boiler (*C.LargeVolBoiler*= TRUE/FALSE):

- Pump is started.

  – Burner is started when water circulation is confirmed (for small capacity boilers, the use of a flow switch is mandatory).
  – Valve is operated depending on the return water temperature.

2) Stopping procedure (*BoilerEnableCommand* = off)

  – Burner is stopped.
  – Valve is shut off.
  – After a pre-defined delay, the pump is stopped.

Boiler Enable

The Boiler Enable signal is activated when the effective *HeatDemand* analogue value gathers information from all the heat consumption blocks and goes above the value of the *C.HeatSystemEnableTemp* parameter. Internal configuration parameters *C.BoilerMinOnTime/C.BoilerMinOffTime* define respective minimum on/off times for the heat generation block. When inactive all the staged and modulated boilers are forced to off.

In order to cover transfer heat loss, *EffHeatDemand* is calculated as a sum of the internal parameters of *HeatDemand* input and *C.HtgSpOffset*.

To prevent rapid temperature changes in the pipes when the heating system is activated, *EffHeatDemand* temperature is ramped up from the current value of the S*ecSupWaterTemp* to obtain *EffSecTempSp*. The internal parameter of *C.SecSpChgRatio* sets the maximum *EffSecTempSp* change per minute.
C.MaxSecWaterTempSp and *C.MinSecWaterTempSp* parameters define limits for the system supply temperature set point.

PI controller compares *EffSecTempSp* with *BoilerWaterTemp* values and calculates *PidBoilerMain* signal which is then used to activate or deactivate boiler stages.

Boiler Valve Operation and Return Water Temperature Control

3-Way Valve Operation

If a boiler is equipped with a 3-way valve and *RetWaterTemp* sensor is connected, the valve opening is controlled by PI controller with the set point defined by *EffRetWaterTempSp*. This feature is used to prevent condensation of vapour inside the boiler. Thus, return water temperature needs to be adjusted in accordance with boiler specification.

C.RetTempCtrlMode parameter defines the control mode.

Control Mode	Description
1 – Control Inactive	Control is not active.
2 – Constant Setpoint	Parameter *C.MaxPrimRetWaterTempSp* is used as set point value.
3 – Boiler Temperature Dependent	Setpoint is calculated using characteristic defined by points *C.LowBoilerTemp* -> *C.MinBoilerRetWaterTempSp*, *C.HighBoilerTemp* -> *C.MaxBoilerRetWaterTempSp.*
4 – No valid value	If the value of *C.RetTempCtrlMode* is not valid, Boiler Temperature Dependent mode will be used.

When *BoilerReturnTemp* drops below *EffRetWaterTempSp*, the PI algorithm limits the maximum valve opening. By default 50% PID bias is applied, meaning that actual PID action starts when *BoilerReturnTemp* drops below *EffRetWaterTempSp* plus ½ PID P_b.

2-Way Valve Operation

If a boiler is equipped with a 2-Way Valve (or *RetWaterTemp* is not available), the valve will be considered as an On/Off shutoff valve. It will be opened to 100% by the start-up procedure and closed during shutdown.

Pump Operation and Alarm

Pumps are started according to the startup procedure described in *Boiler Startup and Stopping*.

Pump status signals can be connected, and depending on the *PumpStatus*, *PumpStartCmdFb* and *PumpFault* parameters, they can generate alarms in different ways such as:

- *PumpFault* is treated as a direct fault signal and the pump alarm will be generated immediately when *PumpFault* input is true.
- *PumpStartCmdFb* and *PumpStatus* are treated as pump operation feedback signals. *PumpStatus* has priority over *PumpStartCmdFb*, and if both are connected, the latter is ignored. If feedback fails to follow *PumpCmd* signal, *RunAlarm* (fail to stop) and *StopAlarm* (fail to start) signals will be generated (with corresponding *C.RunAlmDelay* and *C.StopAlmDelay* times).

An algorithm outputs a *PumpStatus.* This multistate output signal provides a coded pump status, with the following values:

- 1 – Stop
- 2 – Start
- 3 – Run Alarm (fail to stop)
- 4 – Stop Alarm (fail to start)
- 5 – Fault Alarm

If a boiler pump trips into either Stop Alarm or Fault Alarm, the entire boiler will be stopped immediately.

Effective Secondary Supply Water Temperature

The Stand Alone Boiler Control block checks if *SecSupWaterTemp* and *PrimRetWaterTemp* sensors are connected, then chooses higher value to determine the effective supply water temperature, *EffSecSupWaterTemp*.

This procedure is based on the assumption that if *PrimRetWaterTemp* is higher than *SecSupWaterTemp*, concludes that all heat recipient circuit valves are closed and water in the secondary circuit is stagnant which would lead to unreliable information on actual supply temperature from the heat generators.

Time to Reach Setpoint Prediction

The system observes the ratio of change of the *EffSecSupWaterTemp* and compares it with *EffSecTempSp*. It then applies *C.ControlDeadband* to predict the time to reach setpoint. The Control Block will block the on/off switching of the current stage if the predicted time is less than *C.TimeToSwitchStg*. This prevents excessive boiler stage switching.

Active Stage and PI Control

The *ActiveStage* internal value holds information on the current control point of the boiler. Based on the *C.BoilerPowerStage1/2* and if the boiler has a modulated burner, the power ratio between current, previous and next stage in the sequence is calculated. This relation is transferred as *PIDBias* to the main PID to define the regulation of the neutral point.

For modulated boilers, the modulation signal is calculated in parallel with the second stage of each boiler. It climbs from 0% to 100% as the PID goes from *C.ModCtrlOffset* to 100% - *C.ModCtrlOffset*.

When *ActiveStage* changes, the control is locked for a time equal to *C.StgChgDelay.* This control delay gives boilers the necessary time to engage the commanded power. The PID operation is then released and begins from the *PIDBias* point.

If the PID reaches 100% and no other restraints are active, the next stage will be switched on. If on the other hand PID drops to 0% *ActiveStage* will be decreased.

Active Stage Switch Locking

ActiveStage would not be switched if:

- *C.StgChgDelay* time did not elapse after the last change.
- *TimeToReachSp* is smaller than *C.TimeToSwitchStg*.
- Either *SwitchOnDelay* or *SwitchOffDelay* signals are active (applies only to respective On or Off action).
- Signal *StgOvrHeatAct* is active (applies only to On action).

One- or Two-Stage Burner Control

Each of the stages has its own MinOn and MinOff timer to prevent frequent stage switching. There is also a minimum switch on delay for the second stage to avoid direct switching of both stages.

One-Stage Burner

If a boiler is equipped with a one-stage burner (parameter *C.BoilerPowerStage2* = 0). Then stage one will be started when PID goes up to 100% and switched off when PID drops down to 0%.

Two-Stage Burner

If a boiler is equipped with two-stage burner, then both stages will operate with the correct setting of the *PIDBias* as a function of the currently engaged power and power of the next/previous stage.

Modulating Burner Control

Burner Start and Control Command

As far as switching of the stage 1 is concerned, the modulated boiler follows exactly the same principle as a 2-stage boiler. However, when stage 1 is active and PID control is released, the PID regulator will start to modulate burner operation enabling precise *BoilerWaterTemp* regulation.

Boiler Maximum Temperature Protection

Boiler control block has several independent overheat protection mechanisms. They will work correctly only if correct parametrisation is preserved:

- *C.MaxBoilerWaterTempSp* is the lowest value (default 80°C)
- *C.MaxBoilerTempStg2* is next (default 85°C)
- *C.MaxBoilerTempStg1* is next (default 90°C)
- Mechanical HighTempThermostat should be set to value no less than *C.MaxBoilerTempStg1* + 10°C.

Please note the maximum permissible boiler and installation water temperature, and adjust all the values accordingly starting downwards from the highest (set point of the *HighTempThermostat*).

As *BoilerTemp* increases and goes beyond values defined by *C.MaxBoilerWaterTempSp*, PID will start decreasing its control signal. However, if the temperature still rises and reaches *C.MaxBoilerTempStg2*, second stage is immediately switched off. If temperature still rises and goes beyond *C.MaxBoilerTempStg1* first stage is switched off. If temperature continues rising and reaches *C.MaxBoilerTempStg1* + 5°C, the *BoilerOverheat* override state is initiated.

The Boiler Overheat emergency state forces the start of all consumer circuits together with their pumps and sets their temperature setpoints to maximum values. This allows fast heat dissipation from the overheated boiler. In this state (similarly to *HighTempAlarm* state), burner stages/modulation will be disabled, valves stay 100% open and pumps continue running.

If the temperature drops below respective setpoints and hysteresis values (default hysteresis is equal to 5°K, except for *BoilerOverheat* = 10°K), boiler will restart normal operation.

An exception to this rule is activation of the *HighTempAlarm*. If this alarm is tripped, it requires a manual reset. This is why correct adjustment of all the protection thresholds is essential.

Critical Boiler Alarms

If any of the following alarms are activated, the boiler will be switched off immediately, with special operation of the pump and valve in case of High Temperature Alarm.

Valve Failed to Open Alarm

If *IsolationValveSwitch* signal is connected, valve opening command was issued and there was no confirmation of valve opening before the alarm timer elapsed, *ValveClosedAlarm* will be generated and boiler and pump will be stopped immediately.

Pump Alarm

If a boiler pump goes into alarm mode causing it to stop, the boiler will also be switched off immediately.

Low Water Level Alarm

If *LowBoilerWaterLev* signal is connected and it shows that there is no water in the system, *LowWater-LevelAlarm* will be generated, and boiler and pump will be stopped immediately.

High Temperature Alarm

If *HighTempThermostat* signal is activated, HighTempAlarm will be generated and the boiler will be stopped immediately. However, the pump will continue operating and valve will be kept open while *HighTempThermostat* stays active to dissipate excessive boiler heat. When the thermostat goes back to normal, the pump will be switched off after the standard delay and the valve will close.

Boiler Alarm

When *BoilerAlarm* input is activated, *BoilerAlarm* will be generated causing the boiler and pump to be stopped immediately.

Boiler Block Cooperation with Heat Generation Control Block

Even though this boiler control block is designed to be used as standalone, it is equipped with all the necessary inputs enabling it to work as subordinate of Heat Generation Control block. This way, in case of an upgrade from a single to a multiple boiler installation, only minor code corrections are needed.

On the input side, interface signals include:

Input Interface Signal	Description
ExtBoilerCmdStg1/2	Binary inputs, which when connected, will overwrite internal temperature control driven boiler stage commands.
ExtBoilerModCtrl	Analogue input, which when connected, will overwrite internal modulation commands.

On the output side, interface signals include:

Output Interface Signal	Description
StgOvrHeatAct	Binary output signaling that the boiler is overheated. It prevents Heat Group Control block from activating the higher heating stages.
SwitchOn/OffDelay	Binary output signaling that boiler stages are locked by internal MinOn/MinOff timers. It prevents Heat Group Control block from activating the higher heating stages.

Heat Generation Control Block (2-stage/modulated boilers) (HTG_HeatingGroupPowerBased)

The Heat Generation Control block is activated when the effective *HeatDemand* analogue value gathers information from all the heat consumption blocks and goes above the value of the *C.HeatSystemEnableTemp* parameter.With this information, the block takes care of the optimal sequencing of assigned heat generation components, optimum start and stop, and switching of heat generators.

Inputs

The Heat Generation Control Block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Secondary Supply Water Temperature	Heat consumer supply water temperature (AI).
Heat Demand	Signal combining Heat Demand (°C) from all the heat consumer blocks (AI).
Boiler 1/2 Alarm	Malfunction of specified boiler (BI).
Stage Overheat Active	Feedback signal from boiler blocks with information of overheated boilers (BI).
SwitchOffDelay / SwitchOnDelay	Information from subordinate blocks that MinOff/MinOn timer for heat stages are active (BI).
C.Boiler 1/2 Power Stage 1/2	Mandatory configuration parameter, defining nominal power output of all the boiler stages (AI).
Enable	Activation of the heat generation when the *HeatDemand* analogue value is above the value of the *C.HeatSystemEnableTemp* parameter (BI).

Optional Inputs

Input Parameter	Description
Primary Return Water Temperature	Value of the water temperature at the return to heat generators (AI).
C.Min/Max Secondary Water Temperature Set point	Maximum/minimum heat consumer supply water temperature limitation (AV).
C.Boiler 1/2 Modulated	Configuration parameter defining the enabling of modulation of the corresponding boiler (BI).

Outputs

Listed below are the Heating Circuit block output signals (hardware and logic).

Output Parameter	Description
Effective Secondary Temp Set Point	Effective secondary water temperature set point value (AO).
Effective Heat Demand	Effective consumer heat demand value (AO).
Active Stage	Active control stage of the group controller (AO).
Boiler 1/2 Command Stage 1/2	Boiler stage command (BO).
Boiler 1/2 Control	Boiler modulation control (BO).

Block Functions

Heat Generation Enable

This block is activated when the *HeatDemand* analogue value gathers information from all the heat consumption blocks goes above the value of the *C.HeatSystemEnableTemp* parameter. Internal configuration parameters, *C.GroupMinOnTime/C.GroupMinOffTime,* define respective minumum on/off times for the heat generation block. When inactive, ActiveStage value is forced to 0, thus setting all the stage commands to off.

The leading boiler is changed based on runtime. Internal parameter *C.Runtime* defines the operation period after which the leading boiler change will occur. If any of the boilers drop into alarm, the next boiler in line will take over immediately as a leading boiler.

In order to cover transfer heat losses, *EffHeatDemand* is calculated as a sum of *HeatDemand* input and *C.HtgSpOffset* internal parameter.

To prevent rapid temperature changes in the pipes when heating system is activated, *EffHeatDemand* temperature is ramped up from current value of the *SecSupWaterTemp*, to obtain *EffSecTempSp*. Internal parameter *C.SecSpChgRatio* sets maximum *EffSecTempSp* change per minute. *C.MaxSecWaterTempSp* and *C.MinSecWaterTempSp* parameters define limits for the system supply temperature set point.

The PI controller compares *EffSecTempSp*with *SecSupWaterTemp* value and calculates *PidBoilerCorrection* signal which is then used to activate or deactivate boiler stages.

Boiler Sequence Calculation

Based on mandatory configuration parameters, *C.Boiler1/2PowerStage1/2* block calculates optimal sequence of the available boilers.

Two-stage non-modulated burner

Boiler Sequence Parameter	Description
C.BoilerXPowerStage1	Set as the first stage power output.
C.BoilerXPowerStage2	Set as the second stage power output.

One-stage non-modulated burner

Boiler Sequence Parameter	Description
C.BoilerXPowerStage1	Set as the boiler power output.
C.BoilerXPowerStage2	Set to 0.

Modulated burner

Boiler Sequence Parameter	Description
C.BoilerXPowerStage1	Set as the minimum power output (which cannot be modulated).
C.BoilerXPowerStage2	Set to modulated power output.
C.BoilerXModulated	This optional parameter is set to TRUE, to enable modulation of the corresponding boiler.

Based on the available configuration and value of the internal configuration parameter *C.SkipStep3Active*, the boiler sequence is defined as in following example.

For a two boiler system (e.g. boiler 1 – stage1 70kW, stage2 30kW, boiler 2 – stage1 70kW and stage2 30kW) the switching sequence is (sequence stage numbers are held by *ActiveStege* variable):

C.SkipStep3Active = FALSE

- **On Sequence:**

  1) B1S1 = 70kW
  2) B1S1 + B1S2 = 100kW
  3) B1S1 + B2S1 =140kW
  4) B1S1 + B1S2 + B2S1 = 170kW
  5) B1S1 + B1S2 + B2S1 + B2S2 = 200kW

- **Off Sequence:**

  5) B1S1 + B1S2 + B2S1 + B2S2 = 200kW
  4) B1S1 + B1S2 + B2S1 = 170kW
  3) B1S1 + B2S1 =140kW
  2) B1S1 + B1S2 = 100kW
  1) B1S1 = 70kW

C.SkipStep3Active = TRUE

- **On sequence:**

  1) B1S1 = 70kW
  2) B1S1 + B1S2 = 100kW
  4) B1S1 + B1S2 + B2S1 = 170kW
  5) B1S1 + B1S2 + B2S1 + B2S2 = 200kW

- **Off sequence:**

  5) B1S1 + B1S2 + B2S1 + B2S2 = 200kW
  4) B1S1 + B1S2 + B2S1 = 170kW
  2) B1S1 + B1S2 = 100kW
  1) B1S1 = 70kW

Generally, *C.SkipStep3Active* should be set to FALSE if:

- Two-stage non-modulated boilers are in use.
- Second stage of both boilers is less than 40% of the boiler power output.
- Second boiler can be brought on-line when first boiler is not working at its full capacity.

Effective Secondary Supply Water Temperature

This function checks if *SecSupWaterTemp* and *PrimRetWaterTemp* sensors are connected, then chooses higher value to determine effective supply water temperature *EffSecSupWaterTemp*.

This procedure is based on the assumption that if *PrimRetWaterTemp*is higher than *SecSupWaterTemp*, then it concludes that all heat recipient circuit valves are closed and water in the secondary circuit is stagnant. Therefore, this procedure cannot give reliable information on actual supply temperature from the heat generators.

Time to Reach Setpoint Prediction

The system observes the ratio of change of the *SecSupWaterTemp* and compares it with *EffSecTempSp* applying *C.ControlDeadband* to predict Time To Reach Setpoint. If this predicted time is less than *C.TimeToSwitchStg*, the control block will prevent switching on/off of the next stage of the same boiler. If the predicted time is smaller than *C.TimeToSwitchBoiler*, the control block will prevent switching on/off of the next boiler.

Active Stage and PI Control

ActiveStage internal value holds information on the current control point of the heating system (see *Boiler Sequence Calculation*).

Based on *ActiveStage* value and the *C.Boiler1/2PowerStage1/2* power ratio between current, the previous and the next stage in the sequence is calculated. This relation is transferred as PIDBias to the main PID to define the regulation neutral point.

After *ActiveStage* changes, PID operation is held for the time of *C.StgChgDelay* to give boilers the time necessary to engage the commanded power. When the PID operation is released, it starts from the PIDBias point. If PID reaches 100% and no other restraints are present, the next stage would be switched on. If on the other hand PID drops to 0%, then the previous stage would be engaged.

Boiler modulation signal is calculated in parallel with the second stage of each boiler. If *C.BoilerXModulated* parameter indicates that the operated boiler is equipped with a modulated burner, then *BoilerXCtrl* signal climbs from 0%-100% as PID goes from *C.ModCtrlOffset* to 100% - *C.ModCtrlOffset*.

If *HeatingGroupControl* block is disabled, *ActiveStage* is immediately reset to 0.

Active Stage Switching and Locking

ActiveStage would not be switched if:

- *C.StgChgDelay* did not elapse after the last change.
- *TimeToReachSp* is smaller than *C.TimeToSwitchBoiler* (when switching to the next/previous boiler) or *C.TimeToSwitchStg* (when switching to the next/previous stage of the same boiler).

- Any of the boiler signals *SwitchOnDelay* or *SwitchOffDelay* are active (applies only to respective On or Off action).
- Any of the boilers signals *StgOvrHeatAct* (applies only to On action)

Boiler Control

HeatingGroupControl block sends commands to engage boiler stages and modulation via its *Boiler1/2CmdStg1/2* (boiler stage command) and *Boiler1/2Ctrl* (boiler modulation) control signals.

However, it does not take into account all of the boiler operation restrictions, thus it should not be used to directly operate physical IO signals connected to boiler/burner. Instead, *HTG_BoilerSlave* blocks should be used.

Slave Boiler Control Block (HTG_SlaveBoiler)

The Slave Boiler Control block is similar in operation to the standalone boiler block, but it does not provide a supply water temperature control algorithm. It accepts stage command and modulation control from master *HeatingGroupControl* block and applies all the safety and function driven requirements dependent on the boiler type.

Type	Feature	Description
Boiler Type	Standard	Requires return water temperature high enough to avoid vapour condensation.
	Condensation	Requires low return water temperature in order to achieve condensation and boost efficiency.
Fuel Type	Gas	Usually from a supervisory controller perspective, gas and oil boilers are operated in a similar way. Its built-in burner controller provides a basic sequence of burner operations.
	Oil	
	Solid Fuel (generally not used in the automated boiler rooms)	
Burner Type	Stage burners	Usually 1 or 2 stages.
	Modulated burners	
Size (water capacity)	Large Boilers	Can be started with valves closed and pump not running, and starts circulation after water temperature reaches a minimum value.
	Small Boilers	Cannot be started before water circulation is confirmed.

Usually boilers and burners are equipped with dedicated control systems that provide basic control and all the necessary security functions. Boilers, which have the option of external control, must be supplied with a build-in controller able to accept commands from external control systems and send alarms and confirmations. This approach usually relieves the automation system programmer from providing low level burner sequence programming.

Slave boiler control block can be configured to operate staged on/off (1 or 2 stages) or modulated boilers.

Inputs

The Slave Boiler Control block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
External Boiler Command Stage1	External command to start first stage of the boiler (BI)
Boiler Water Temperature Sensor	Temperature value of the water at the boiler output (AI)
Heat Demand	Supply water temperature request from the heat consumer blocks (AI)
Boiler Alarm	Burner or boiler internal fault signal, active state results in immediate boiler switch off (BI).
High Temperature Thermostat	One or two stage thermostat that is used to stop boiler operation in case temperature inside of the boiler rises too much. If thermostat has two stages, the first stage is auto reset and the second stage requires manual reset directly on the boiler (BI)
Reset	Alarm reset command (BI)

Optional Inputs

Input Parameter	Description
External Boiler Command Stage2	If connected, used to start second stage of the boiler (BI)
External Boiler Modulated Control	If connected used to control burner modulation (AI)
Boiler Return Water Temperature Sensor	Temperature value of the water entering the boiler (AI)
Boiler Pump Status	Confirms operation of the pump (feedback from a contactor or a flow switch). Do not use simultaneously with Pump Start Command Feedback (BI)
Boiler Pump Command Feedback	Confirmation signal from the output of the controller that it is energized. Do not use simultaneously with Pump Start Command Feedback (BI)
Boiler Pump Fault	Fault signal of the pump (feedback from a contactor or a flow switch) (BI)
Low Water Level Sensor	Fault signal used to block boiler operation if there is no water in the system (BI)
Isolation Valve Switch	Feedback signal from 2-way isolation valve, used to confirm valve opening (BI)
Pump Exercise	Activation command for periodic pump and valve motorisation (BI)
Boiler Enable	Boiler start signal. Might be omitted if *ExtBoilerCmdStage1* is used (BI)

Outputs

Input Parameter	Description
Boiler Command Stage 1/2	Boiler stage enable output (BO)
Boiler Modulation Control	Boiler modulation control output (AO)
Boiler Valve Control	Boiler control/isolation valve output (AO)
Boiler Pump Command	Burner or boiler internal fault signal, active state results in immediate boiler switch off (BI)
Boiler Pump Status	Multistate pump status point (MO)
Boiler Pump Run Alarm	Fail to stop alarm active (BO)
Boiler Pump Stop Alarm	Fail to start alarm active (BO)
Boiler Pump Fault Alarm	Direct fault alarm active (BO)
Boiler Alarm	Direct burner/boiler fault alarm active (BO)
High Temperature Alarm	*HighTempThermostat* trip alarm active (BO)
Low Water Level Alarm	Low water level alarm active (BO)
Valve Closed Alarm	Isolation valve failed to open alarm active (BO)
Boiler Overheat	Boiler overheat priority signal (BO)
Stage Overheat Active	Boiler stages are switched off due to too high value of the *BoilerWaterTemp.* Used if block is used as a subordinate for group control block. (BO)
Stage On/Off Delay Active	Boiler stages change is locked by internal on/off delay. Used if block is used as a subordinate for group control block. (BO)
Effective Return Water Temp Set Point	Effective boiler return water temperature set point value (AO)

Block Functions

Boiler Startup and Stopping

When this block receives the *BoilerEnable*command (optionally if *ExtBoilerCmdStg1*is activated), it activates the startup procedure if there are no alarm conditions active.

For 2-way valve (or no valve) (*C.3WayValve* = FALSE)

1) Starting large water capacity boiler (*C.LargeVolBoiler*= TRUE):

- Burner is started.
- When boiler temperature reaches defined temperature, valve is opened.
- When valve is opened pump is started.

2) Starting small water capacity boiler (*C.LargeVolBoiler*= FALSE):

– Valve is opened.
– When valve is opened (confirmed by valve end switch) pump is started. If end switch is not available, a delay time of 120s is applied.
– When water circulation is confirmed, the burner is started

3) Stopping procedure (*BoilerEnableCommand* = off)

– Burner is stopped.
– After a pre-defined delay, the pump is stopped.

For 3-way valve (*C.3WayValve* = TRUE)

1) Starting large or small water capacity boiler (*C.LargeVolBoiler*= TRUE/FALSE):

– Pump is started.
– Burner is started when water circulation is confirmed (for small capacity boilers, the use of a flow switch is mandatory).
– Valve is operated depending on the return water temperature.

2) Stopping procedure (*BoilerEnableCommand* = off)

– Burner is stopped.
– Valve is shut off.
– After a pre-defined delay, the pump is stopped.

Boiler Valve Operation and Return Water Temperature Control

3-Way Valve Operation

If a boiler is equipped with a 3-way valve and *RetWaterTemp* sensor is connected, the valve opening is controlled by PI controller with the set point defined by *EffRetWaterTempSp*. This feature is used to prevent condensation of vapour inside the boiler. Thus, return water temperature needs to be adjusted in accordance with boiler specification.

C.RetTempCtrlMode parameter defines the control mode.

Control Mode	Description
1 – Control Inactive	Control is not active.
2 – Constant Setpoint	Parameter *C.MaxPrimRetWaterTempSp* is used as set point value.
3 – Boiler Temperature Dependent	Setpoint is calculated using characteristic defined by points *C.LowBoilerTemp* -> *C.MinBoilerRetWaterTempSp*, *C.HighBoilerTemp* -> *C.MaxBoilerRetWaterTempSp.*
4 – No valid value	If the value of *C.RetTempCtrlMode* is not valid, Boiler Temperature Dependent mode will be used.

When *BoilerReturnTemp* drops below *EffRetWaterTempSp*, the PI algorithm limits the maximum valve opening. By default 50% PID bias is applied, meaning that actual PID action starts when *BoilerReturn-Temp* drops below *EffRetWaterTempSp* plus ½ PID P_b.

2-Way Valve Operation

If a boiler is equipped with a 2-Way Valve (or *RetWaterTemp* is not available), the valve will be considered as an On/Off shutoff valve. It will be opened to 100% by the start-up procedure and closed during shutdown.

Pump Operation and Alarm

Pumps are started according to the startup procedure described in Boiler Startup and Stopping.

Pump status signals can be connected, and depending on the *PumpStatus*, *PumpStartCmdFb* and *PumpFault* parameters, they can generate alarms in different ways such as:

- *PumpFault* is treated as a direct fault signal and the pump alarm will be generated immediately when *PumpFault* input is true.

- *PumpStartCmdFb* and *PumpStatus* are treated as pump operation feedback signals. *PumpStatus* has priority over *PumpStartCmdFb*, and if both are connected, the latter is ignored. If feedback fails to follow *PumpCmd* signal, *RunAlarm* (fail to stop) and *StopAlarm* (fail to start) signals will be generated (with corresponding *C.RunAlmDelay* and *C.StopAlmDelay* times).

An algorithm outputs a *PumpStatus.* This multistate output signal provides a coded pump status, with the following values:

- 1 – Stop
- 2 – Start
- 3 – Run Alarm (fail to stop)
- 4 – Stop Alarm (fail to start)
- 5 – Fault Alarm

If a boiler pump trips into either Stop Alarm or Fault Alarm, the entire boiler will be stopped immediately.

One- or Two-Stage Burner Control

Each of the stages has its own MinOn and MinOff timer to prevent frequent stage switching. There is also a minimum switch on delay for the second stage to avoid direct switching of both stages.

Modulating Burner Control

Burner Start and Control Command

As far as switching of the stage 1 is concerned, the modulated boiler follows exactly the same principle as a 2-stage boiler. However, when stage 1 is active and PID control is released, the PID regulator will start to modulate burner operation enabling precise *BoilerWaterTemp* regulation.

Boiler Maximum Temperature Protection

Boiler control block has several independent overheat protection mechanisms. They will work correctly only if correct parametrisation is preserved:

- *C.MaxBoilerTempStg2* is next (default 85°C)
- *C.MaxBoilerTempStg1* is next (default 90°C)
- Mechanical HighTempThermostat should be set to value no less than *C.MaxBoilerTempStg1* + 10°C.

Please note the maximum permissible boiler and installation water temperature, and adjust all the values accordingly starting downwards from the highest (setpoint of the *HighTempThermostat*).

As *BoilerTemp* increases and goes beyond values defined by *C.MaxBoilerTempStg2,* second stage is immediately switched off. If temperature still rises and goes beyond *C.MaxBoilerTempStg1,*first stage is switched off. If temperature keeps on rising and reaches *C.MaxBoilerTempStg1 + 5°C – BoilerOverheat* override state is initiated.

The Boiler Overheat emergency state forces the start of all the consumer circuits together with their pumps and sets their temperature setpoints to maximum values, thus allowing fast heat dissipation from the overheated boiler. In this state (similarly to *HighTempAlarm* state), burner stages/modulation will be disabled, and valve stays 100% open and pump continues operation.

If the temperature drops below respective setpoints and hysteresis values (default hysteresis is equal to 5°K, except for *BoilerOverheat* = 10°K), the boiler will restart normal operation.

An exception to this rule is the activation of the *HighTempAlarm*. If this alarm is tripped, it requires a manual reset. This is why correct adjustment of all the protection thresholds is essential.

Critical Boiler Alarms

If any of the following alarms are activated, the boiler will be switched off immediately, with special operation of the pump and valve in case of High Temperature Alarm.

Valve Failed to Open Alarm

If *IsolationValveSwitch* signal is connected, valve opening command was issued and there was no confirmation of valve opening before the alarm timer elapsed, *ValveClosedAlarm* will be generated and boiler and pump will be stopped immediately.

Pump Alarm

If a boiler pump goes into alarm mode causing it to stop, the boiler will also be switched off immediately.

Low Water Level Alarm

If *LowBoilerWaterLev* signal is connected and it shows that there is no water in the system, *LowWaterLevelAlarm* will be generated, and boiler and pump will be stopped immediately.

High Temperature Alarm

If *HighTempThermostat* signal is activated, HighTempAlarm will be generated and the boiler will be stopped immediately. However, the pump will continue operating and valve will be kept open while *HighTempThermostat* stays active to dissipate excessive boiler heat. When the thermostat goes back to normal, the pump will be switched off after the standard delay and the valve will close.

Boiler Alarm

When *BoilerAlarm* input is activated, *BoilerAlarm* will be generated causing the boiler and pump to be stopped immediately.

Boiler Block Cooperation with Heat Generation Control Block

A Slave Boiler control block is designed to be used as subordinate of Heat Generation Control block. It has all the necessary interface points to enable this cooperation.

On the input side, interface signals include:

Input Interface Signal	Description
ExtBoilerCmdStg1/2	Binary inputs used as direct start commands for their respective stages.
ExtBoilerModCtrl	Analogue input used as modulation command.

On the output side, interface signals include:

Output Interface Signal	Description
StgOvrHeatAct	Binary output signaling that the boiler is overheated. It prevents Heat Group Control block from activating the higher heating stages.
SwitchOn/OffDelay	Binary output signaling that boiler stages are locked by internal MinOn/ MinOff timers. It prevents Heat Group Control block from activating the higher heating stages.

District Heating Systems (HTG_DistrictHeatingStandalone)

District heating solutions are most common in cities where buildings are connected to central heating plants providing thermal energy by distribution of the hot water. In this situation, the temperature of the water supplied from the central heating plant is usually a function of the outside air temperature. The heating distributor often requires that the temperature of the water returned to the central system is kept under a specified value (usually outside air temperature or supply water temperature dependent).

When it comes to placing the district heating in the structure of the heating system, it should be placed in the heat generation level. Some systems might even consist of district heating as the main heating system with an optional small boiler room as a backup system.

In district heating systems, one of the major issues is to provide controllability in winter and summer conditions, when heat consumption, and thus water flow in the primary (central plant side) coil differ greatly. It can be obtained by careful valve sizing (which is always a compromise), or by using two independent valves supplying one heat exchanger, or even two valves and two heat exchangers.

Large buildings usually have multiple heat exchangers that can be assigned to perform different functions. For example one heat exchanger used for AHU heaters, other for radiators or fan coils and another for a domestic hot water system. This approach allows the optimization of heat exchanger and valve sizing, thus having better efficiency and regulation precision.

Inputs

The block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Secondary Supply Water Temperature Sensor	Temperature value of the supply water in the secondary circuit (AI).
Heat Demand	Supply water temperature request from the heat consumer blocks (AI).
Outside Air Temperature	Filtered, average value of the outside air temperature. Filtration time should be between 1-3h (AI).
Enable	Heating system start command (BI).
Reset	Alarm reset command (BI).

Optional Inputs

Input Parameter	Description
High Temperature Thermostat	One or two stage thermostat that is used to stop boiler operation in case the temperature inside of the boiler rises too much. If the thermostat has two stages, the first stage is auto reset and the second stage requires manual reset directly on the boiler (BI).
Primary supply Water Temperature Sensor	Temperature value of the supply water in the primary circuit (AI).
Primary Return Water Temperature Sensor 1/2	Temperature value of the return water in the primary circuit (heat exchanger 1/2) (AI).
Secondary Pump Status	Confirms operation of the pump (feedback from a contactor or a flow switch). Do not use simultaneously with Pump Start Command Feedback (BI).
Secondary Pump Command Feedback	Confirmation signal from the output of the controller that it is energized. Do not use simultaneously with Pump Start Command Feedback (BI).
Secondary Pump Fault	Fault signal of the pump (feedback from a contactor or a flow switch) (BI).
Pump Exercise	Activation command for periodic pump and valve motorisation (BI).

Outputs

Input Parameter	Description
Heat Exchanger Valve Control 1/2	Heat exchanger 1/2 valve control (AO).
Secondary Pump Command	Start signal for primary side pump (BO).
Secondary Pump Status	Multistate pump status point (MO).
Secondary Pump Run Alarm	Fail to stop alarm active (BO).
Secondary Pump Stop Alarm	Fail to start alarm active (BO).
Secondary Pump Fault Alarm	Direct fault alarm active (BO).
High Temperature Alarm	*HighTempThermostat* trip alarm active (BO).
Boiler Overheat	Boiler overheat priority signal (BO).
Effective Heat Demand	Effective heat demand value (AO).
Effective Secondary Water Temp Set Point	Effective supply water temperature set point value (AO).
Active Stage	Information internal operation stage of the boiler (AO).

Block Functions

System Configuration

District heating systems can be divided into four major groups:

Heating System Type	Description
1 – 1 primary valve	Mixing valve located on the return pipe. One *PrimRetWaterTemp* sensor may be present in the system.
2 – 2 primary valves	Two valves connected in parallel (the *C.ValveSizeRatio* parameter defines how large of a percentage of the total control signal span is accorded to the first valve - by default 50%). This provides better control in case of large flows (especially in low load conditions). One *PrimRetWaterTemp* sensor may be present in the system. Leading valve will by periodically changed to provide equal ageing of the valves.
3 – 2 primary heat exchangers	Connected in parallel on primary side and in serial on secondary side. Each of the heat exchangers has one valve and can have one *PrimRetWaterTemp* sensor.
4 – 2 primary valves staged	Two unequally sized valves connected in parallel. One valve has a k_{vs} that is at least twice the size of the other. The smaller valve provides much better controllability in low load conditions. One *PrimRetWaterTemp* sensor may be present in the system.

Multistate configuration parameter *C.SystemType* defines district heating system configuration.

Heat Generation Enable

The block is activated when the effective *HeatDemand* analogue value gathers information from all the heat consumption blocks and goes above the value of the *C.HeatSystemEnableTemp* parameter. Internal configuration parameters *C.GroupMinOnTime/C.GroupMinOffTime* define respective minimum on/off times for the district heating block.

In order to cover transfer heat losses, *EffHeatDemand* is calculated as a sum of *HeatDemand* input and *C.HtgSpOffset* internal parameters.

To prevent rapid temperature changes in the pipes when heating system is activated, *EffHeatDemand* temperature is ramped up from current value of the *SecSupWaterTemp* to obtain *EffSecTempSp*. Internal parameter *C.SecSpChgRatio* sets maximum *EffSecTempSp* change per minute. *C.MaxSecWaterTempSp* and *C.MinSecWaterTempSp* parameters define limits for the system supply temperature set point.

PI controller compares *EffSecTempSp* with *SecSupWaterTemp* value and calculates the *PidDistHeatMain signal* that is then used to control primary side valve/valves and in effect the heat output of the heat exchanger.

Forced Cooldown Protection

When system starts up and *PrimSupWaterTemp* is present, it then runs for 15 minutes. After this time *PrimSupWaterTemp* value is compared to the *SecSupWaterTemp*.

If *PrimSupWaterTemp* is not higher by at least *C.PrimToSecTempDiff,* then valve control will be disabled to prevent secondary circuit forced cooldown.

However, if system is blocked to prevent cooldown, control will be enabled periodically every 300s for 60s to allow water flow through the primary system for reliable measurement of primary water temperature.

District Heating Start-up and Stopping

1) When District Heating block receives *Enable* command, it:

- Starts the pump
- Sets starting point of the *EffSecWaterTempSp* ramp to current value of the *SecSupWaterTemp*
- Enables temperature regulation

2) When *Enable* command is switched off, the block enters Stopping mode, it:

- Stops regulation and closes the valve
- Activates the pump switch off timer
- When time elapses, switches off the pump

Valve Control - 1-Valve

When temperature regulation is enabled, internal PI controller compares *EffSecWaterTempSp* with *SecSupWaterTemp* and calculates control signal with respect to raise ramp defined by the *C.ValveRampTime* (which limits maximum control signal raise per minute). The valve will not open until its control signal goes above *C.MinValveOpening* parameter (to prevent high cavitation noise on the almost fully closed valve).

Valve Control - 2-Valve Sequential Control

If two valves are used, and they have similar size (k_{vs} ratio from 1/1 to 2/1), sequential control is used. If valves have different k_{vs} parameters, the smaller valve should be always valve 1.

When temperature regulation is enabled, the internal PI controller compares *EffSecWaterTempSp* with *SecSupWaterTemp* and calculates the control signal. This signal is then divided between two valves. The first valve operates from 0% of the PID control signal to *C.ValveSizeRatio*. The second valve starts to operate when the PID signal reaches *C.ValveSizeRatio-C.ValveOverlap* until 100%. This ensures much better regulation precision, especially with small loads, when the use of just one big valve would not allow to stabilise the supply water temperature.

The first valve will not open until its control signal goes above *C.MinValveOpening* parameter (to prevent high cavitation noise on the almost fully closed valve).

Valve Control - 2 Staged Ramp

If two valves are used, and they differ significantly in the size, staged ramp function should be applied. First, the number of steps must be calculated and should be equal to:

[N_{steps} = C.ValveKvsRatio + 1], where [C.ValveKvsRatio = (Big valve k_{vs} / Small valve k_{vs})].

When regulation starts, the smaller valve is operated from 0% to 100% when the valve demand signal goes from 0% to [100% / N_{steps}]. If the demand goes higher, then *ActiveStage* is increased and the larger valve opening is set to the value of [(Big valve k_{vs} / Small valve k_{vs}) × *ActiveStage*], while simultaneously, the smaller valve is set to 0%. This sequence is repeated until both valves reach 100% opening.

When heat demand decreases, the smaller valve is gradually shut. When it reaches 0%, the control signal has to drop by [C.ValveOverlap / N_{steps} + 1] more for the *ActiveStage* to be decreased and the larger valve being shut by one step. This provides the necessary hysteresis, and in general, allows very precise control in a very wide range of water flow rates.

C.MinValveOpening parameter is not used in this mode.

Primary Coil Return Water Temperature Control

If *PrimeRetWaterTemp1/2*sensor is present, it can be used to limit the temperature of water returned to the district heating network.

ReturnTempControlMode parameter defines control mode:

Control Mode	Description
1 – Control inactive	Control is not active.
2 – Constant Setpoint	Parameter *C.MaxPrimRetWaterTempSp* is used as setpoint value.
3 – Primary Supply Temperature dependent	Setpoint is calculated using characteristic defined by points *C.LowPrimSupTemp -> C.MinPrimRetWaterTempSp*, *C.HighPrimSupTemp -> C.MaxPrimRetWaterTempSp*
4 – Outside Air Temperature dependent	Setpoint is calculated using characteristics defined by points *MinPrimReturnReference -> MinPrimReturnSetpoint*, *MaxPrimReturnReference -> MaxPrimReturnSetpoint*, and *OutsideAirTemp* is used as reference value.

Primary water return temperature will be activated 300s after valves are opened (control signal climbs to a value higher than 10%), and will be deactivated 30s after they are closed (control signal drops below 2%). This ensures that the return water temperature regulation algorithm would not refer to the temperature of stagnant water. When regulation is activated, PID algorithm checks if *PrimRetWaterTemp1/2* is higher than *EffRetWaterTempSp* and then limits the maximum value of the heat demand signal transferred to the heating valves.

PrimRetWaterTemp2 will only be used to limit valve 2 operation in case heating system is configured as "*2PrimaryHeatExchangers*".

Pump Operation and Alarm

Pumps are started according to the startup procedure described in *District Heating Start-up and Stopping*.

Pump status signals can be connected, and depending on the *PumpStatus*, *PumpStartCmdFb* and *PumpFault* parameters, they can generate alarms in different ways such as:

- *PumpFault* is treated as a direct fault signal and the pump alarm will be generated immediately when *PumpFault* input is true.
- *PumpStartCmdFb* and *PumpStatus* are treated as pump operation feedback signals. *PumpStatus* has priority over *PumpStartCmdFb*, and if both are connected, the latter is ignored. If feedback fails to follow *PumpCmd* signal, *RunAlarm* (fail to stop) and *StopAlarm* (fail to start) signals will be generated (with corresponding *C.RunAlmDelay* and *C.StopAlmDelay* times).

An algorithm outputs a *PumpStatus.* This multistate output signal provides a coded pump status, with the following values:

- 1 – Stop
- 2 – Start
- 3 – Run Alarm (fail to stop)
- 4 – Stop Alarm (fail to start)
- 5 – Fault Alarm

High Temperature Alarm

If *HighTempThermostat* signal is activated, *HighTempAlarm* will be generated and *OverheatOverride* is activated.

OverheatOverride emergency state forces the start of all the consumer circuits together with their pumps and sets their temperature setpoints to maximum values, thus allowing fast heat dissipation from the overheated supply system. In this state, valve/valves will be shut immediately and pumps will be forced to operate while *HighTempThermostat* stays active. When the *HighTempThermostat* signal deactivates, *OverheatOverride* will be switched off after a 60 second delay.

Heat Distriburtion

Heat distribution blocks create a link between Heat Generation and Heat Consumption blocks.
Main classes of the blocks at this control level are:

Block Type	Description
Heating Generation Control blocks	These blocks are responsible for heat generator sequencing, alarm switching and supervisory control based on the Heat Demand signal (for block functionality refer to Application Library boiler sequence block).
Heat Demand Calculation block	This block is responsible for determining the beginning and the end of the heating season, gathering heat demands from Heat Consumption block, calculation of the heating curve value and passing those values to other blocks.
Additional blocks	Double pump switch, heating curve calculation, heat demand, coil pressure control, etc.

Logarithmic Heating Curve (HTG_HeatingCurve)

Heating Curve block adjusts heat demand in function of the Outside Air Temp. It can be parametrised using Heating Curve type, corresponding to heating curves defined by the EN 12098-1 standard, or by direct setting characteristic parameters.

Inputs

The Heating Curve Logarithmic Block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Average Outside Air Temperature	Filtered, average value of the outside air temperature. Filtration time should be between 1-3h. (AI)

Optional Inputs

Input Parameter	Description
Heating Curve Offset	Additional correction which is added to calculated value of the Heating Curve. (AI)
TempOffset	A value at which Heating Curve equals Outside Air Temp. Default value 20°C. (AI)
Slope	Decides the shape of the heating curve (its steepness). The higher value the steeper is the characteristic. Default value 1.5. (AI)
Ramp Ratio	Enables *HeatingCurve* ramping, to avoid problems emerging from excessively fast thermal expansion of the pipes. Default value 1 °C/min (15°/h). (AI)
Min / Max Limit	Limits the *HeatingCurve* output value. Default 10°C/85°C. (AI)
Enable	If the block is disabled, *HeatingCurve* output is equal to the value of *MinLimit*. (BI)

Outputs

Heating Curve Logarithmic Block output signals (hardware and logic).

Output Parameter	Description
HeatingCurve	Effective *HeatingCurve* value output. (AO)
HC_Enabled	Start signal for the pumps. (BO)

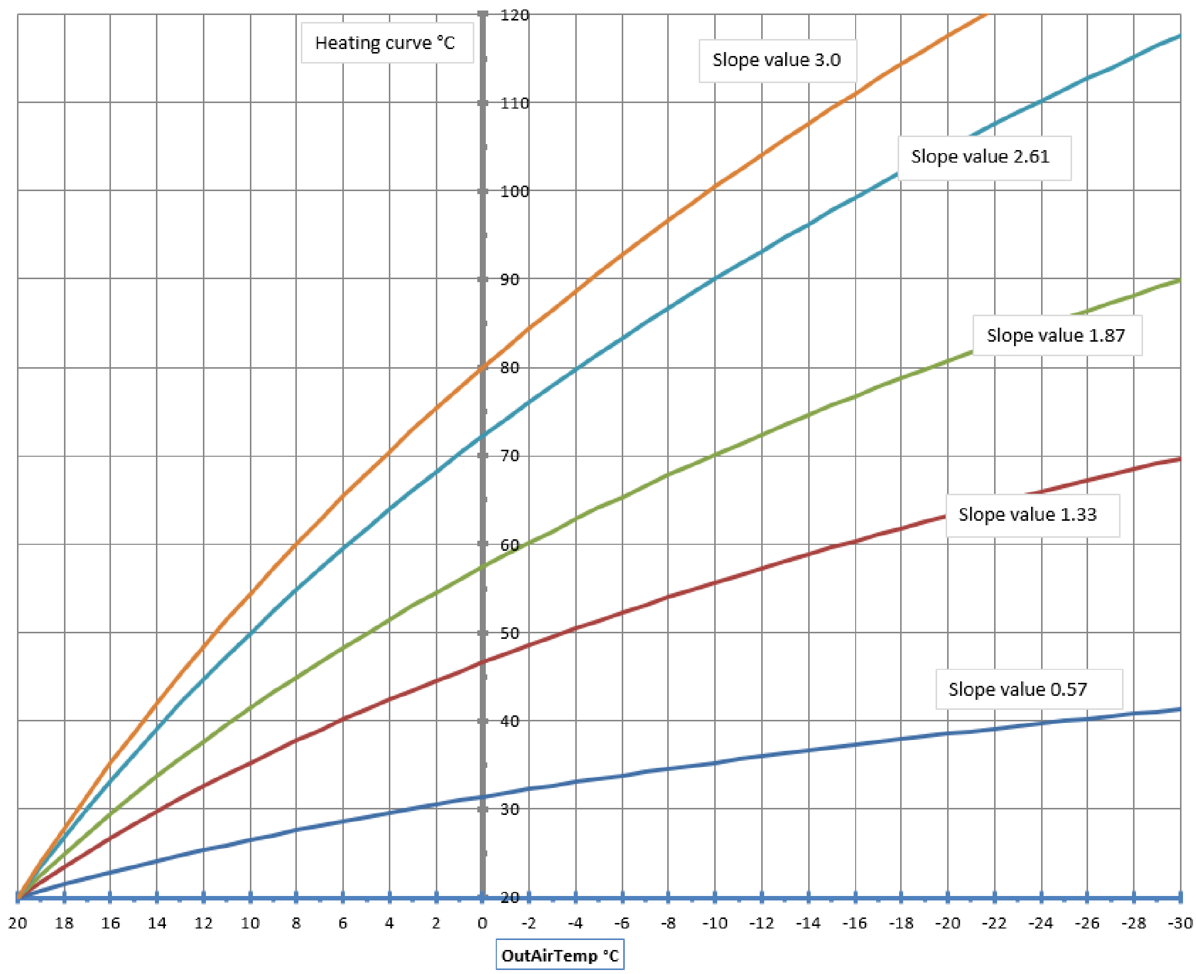
Block Functions

HeatingCurve Calculation Principles

HeatingCurve is defined by *TempOffset* and *Slope. TempOffset* is a value at which *HeatingCurve* is equal to *OutsideAirTemp*. *Slope* defines the steepness of the characteristic. Since the heat exchange ratio between heat emitters and surrounding air increases with the corresponding temperature difference, *HeatingCurve* characteristic is not linear. It gradually flatters out with the heating medium temperature increase.

Slope value should depend on the used heating system type. It should be:

- 0.3 to 0.8 – low temperature system, (e.g. low temperature radiator system, radiant floor heating).
- 0.8 to 1.5 – medium temperature systems (e.g. standard radiator system).
- Above 1.5 – high temperature systems (e.g. fan coils, air handling units etc.).



Predefined HeatingCurve Characteristics

Following EN 12098-1 standard requirements, four predefined *HeatingCurve* characteristics are included in the block. They can be applied by setting *C.HCType* parameter value to:

Parameter	Description
HCType 1 – High temperature heating systems	High temperature AHUs, high temperature radiant heating systems (90°C/70°C water supply/return temperature at -10°C Outside Air Temperature). Corresponding slope value is set by *C.SlopeHCType1* (default value 2.61).
HCType 2 – Medium-high temperature heating systems	Standard temperature AHUs, fan coil systems (70°C/55°C water supply/return temperature at -10°C Outside Air Temperature). Corresponding slope value is set by *C.SlopeHCType2* (default value 1.87).
HCType 3 – Medium temperature heating systems	Standard temperature radiator systems (55°C/45°C water supply/return temperature at -10°C Outside Air Temperature). Corresponding slope value is set by *C.SlopeHCType3* (default value 1.33).
HCType 4 – low temperature heating systems	Radiant floor, low temperature radiator system (35°C /28°C water supply/ return at -10°C Outside Air Temperature). Corresponding slope value is set by *C.SlopeHCType4* (default value 0.57).

Max/Min Limit, HCOffset and RampRatio

After HeatingCurve calculation, *HCOffset* value is added. This allows for easy incorporation of additional correction mechanisms which are designed to fine tune the *HeatingCurve* characteristic (for example room temperature compensation mechanism). The resulting value is then limited by *Max/Min-Limit* to prevent the system from overheating.

The final step is introduction of time based ratio block, which limits *HeatingCurve* output change of value ratio, to prevent problems emerging from excessively fast thermal expansion of the pipes. *RampRatio* parameter defines increase/decrease rate in °C/min.

Linear/Segmented Heating Curve (HTG_HeatingCurveLinear)

Heating Curve linear calculation block adjusts heat demand in function of the Outside Air Temp. It adopts definition of the *HeatingCurve* with use of characteristic points rather than logarithmic formula used in previous block.

Inputs

The Heating Curve Linear Calculation Block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Average Outside Air Temperature	Filtered, average value of the outside air temperature. Filtration time should be between 1-3h. (AI)
OAT_1/HC_1 OAT_2/HC_2	Two points are necessary to define a basic characteristic. (AI)

Optional Inputs

Input Parameter	Description
OAT_3/HC_3 OAT_4/HC_4 OAT_5/HC_5	Additional optional points, which allow definition of more complicated characteristics. Default value *null*. (AI)
Ramp Ratio	Enables *HeatingCurve* ramping, to avoid problems emerging from excessively fast thermal expansion of the pipes. Default value 1 °C/min (15°/h). (AI)
Enable	If the block is disabled, *HeatingCurve* output is equal to the value of *MinLimit*. (BI)

Outputs

Heating Curve Linear Calculation Block output signals (hardware and logic).

Output Parameter	Description
HeatingCurve	Effective *HeatingCurve* value output. (AO)
HC_Enabled	Start signal for the pumps. (BO)

Block Functions

HeatingCurve Calculation Principles

Heating curve is defined by minimum 2 and maximum 5 points. Each of them consists of *OutsideAirTemp* (*Out_1-5*) and the corresponding *HeatingCurve* (*HC_1-5*) values. Beginning and ending points are used to set maximum and minimum limitations for the heat demand in case *OutsideAirTemp* is out of the defined range. Values in between follow linear segments defined by two neighbouring points.

OutsideAirTemp values for the points have to be arranged in descending order, otherwise a conflicting point will be discarded in the calculations.

RampRatio

After HeatingCurve calculation, time based ratio is applied. It limits *HeatingCurve* output change of value ratio to prevent problems emerging from excessively fast thermal expansion of the pipes. The *RampRatio* parameter defines the rate of increase/decrease in °C/min.

Lead Pump Sequencing (GN_LeadPumpSequence)

The Lead Pump Sequencing Block provides periodic rotation of a leading pump, or other equipment, in a group consisting of up to 5 elements (minimum 2 are obligatory).

Inputs

The Lead Pump Sequencing Block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Start Command	Points enabling operation of the block. (BI)
Fault1/Fault2	Direct fault signal. (BI)
Reset	Alarm reset command. (BI)

Optional Inputs

Input Parameter	Description
Fault3/Fault4/Fault5	Direct fault signal for optional pumps. Connection of a valid binary signal enables operation of the corresponding pump, including it in the sequence (BI)
Flow Switch	Water flow confirmation (or other operation feedback), common for all the commanded elements (BI)

Outputs

Lead Pump Sequencing Block output signals (hardware and logic).

Output Parameter	Description
Cmd1/Cmd2/Cmd3/Cmd4/Cmd5	Start signal for the pumps. (BO)
Alarm1/Alarm2/Alarm3/Alarm4/Alarm5	Pump alarm direct fault or fail to start. (BO)

Block Functions

Pump Switching Mode

Parameter *C.SwitchMode* defines a manner in which pump sequencing occurs:

Parameter	Description
1 – RuntimeBased	This block calculates the running time of the active pump. When it exceeds the value defined by the parameter *C.Runtime,* the active pump is switched off and next one in the sequence is started.
2 – DateBased	Switching occurs on preferred weekday, hour and minute (*C.SwitchDay, C.SwitchHour, C.SwitchMinute*). *C.SwitchDay* parameter is binary coded, (for ex. value 31 = 0b0011111 means Monday to Friday, and default value 32 = 0b010000 means Saturday). It allows setting the switching to the preferred day and time, for example when maintenance crew is present, so they can react immediately to the potential malfunction.
3 – DateAndRuntimeBased	This block calculates the pump runtime just as in mode 1, but waits to switch until *SwitchingWeekday* (binary coded) and *SwitchingTime* are reached.

The block will apply switching delay *C.SwitchDelay*. If the value is positive then the next element in the sequence will be started with a delay equal to the *C.SwitchDelay* parameter, while the current one is immediately switched off, thus giving time gap between operation periods of the two elements. If the value is negative, the next element will be switched on immediately, while the previous one will be switched off with a delay equal to the absolute value of the *C.SwitchDelay* parameter, thus resulting in overlapping of the working periods of the two elements.

Pump Alarm Handling

If *Fault* input is activated for a specified pump, it will activate the alarm for this pump.

Pumps for which the Fault signal is not connected will be considered as inactive.

If *FlowSw* input is connected (not equal to *null*), and after sending the start command to a pump, the block will be observing the status of the *FlowSw* input. If it does not provide feedback on pump operation after a delay defined by *C.FlowSwDelay*, the pump alarm will be activated.

When the alarm of the active pump is detected, the next pump is switched on (with the respect to the positive *C.SwitchDelay* time, negative delay value will lead to immediate pump switching).

Pumps with active alarms are excluded from the operation sequence.

Heat Consumption

Heat consumption blocks are responsible for adjusting water supplied by the network to specific needs of the circuits that are connected to distribution system as heat consumers. Typically 3 types of blocks might be used.

Block Type	Description
Heating circuit with variable supply temperature	Typically used for radiators, fan coils, floor heating systems etc. Supply water temperature setpoint is Outside Air Temperature dependent.
Heating circuit with constant supply temperature	Used for installations which need the constant water temperature independently from Outside Air Temperature value. Those might be technological heat (for example steam generator, or technological industrial installations) or specific HVAC installations (for example reheaters, or heaters in dehumidifiers).
Domestic hot water installations	Circuits providing hot water for direct human consumption or hygienic use.

Heating Circuit Block (HTG_HeatingCircuit)

The Heating Circuit Block is the most commonly used type of block in heating systems. It can be used for radiators, floor heating, fan coils, AHU heaters etc. The general idea is that Secondary Supply Water Temperature is adjusted depending on the Outside Air Temperature (by use of the Heating Curve). The Heating Curve has to correspond to the thermal characteristic of the heat emitter (a different characteristic would be used for high temperature radiator, low temperature radiator and for floor heating systems).

The same block is used for the variable temperature and the constant temperature heating circuit. In the second case, Heating Curve input is connected to a constant value setpoint.

Inputs

The Heating Circuit Block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Heat Curve	Value of the heating setpoint calculated by corresponding Heat Curve.
Mixed Water Temperature	Mixed water temperature sensor (AI).
Night Setback Offset	Water temperature difference by which the water temperature setpoint is decreased when night setback function is active (AI).
Night Setback Enable	Point activating night setback function (BI).
Enable	Heating circuit activation (BI).
Reset	Alarm reset command (BI).

Optional Inputs

Input Parameter	Description
High Temperature Thermostat	thermostat that is used to immediately cut off valve in case of circuit overheating (BI).
Return Water Temperature sensor	Used to prevent forced coil cool down (AI)
Room Temperature	Used to correct heating curve depending on the room temperature (AI).
Room Temperature Set Point	Reference value for the room temperature based correction (AI).
Secondary Supply Water Temperature	Value of the water temperature at the supply of the main heating coil (AI).
Primary Return Water Temperature	Value of the water temperature at the return to heat generators (AI).
Pump Status	Confirms operation of the pump (feedback from a contactor or a flow switch). Do not use simultaneously with Pump Start Command Feedback (BI).
Pump Start Command Feedback	Confirmation signal from the output of the controller that it is energized. Do not use simultaneously with Pump Start Command Feedback (BI).
Pump Fault	Fault signal of the pump (feedback from a contactor or a flow switch) (BI).
MinLimit/MaxLimit	Values setting the boundaries for the Effective Water Temperature Set Point. Default 10°C/85°C (AI).
Heating Valve Feedback	Feedback information from Heating Valve output (AI).
Boiler Overheat	Boiler overheat priority signal (BI).
DHW Priority	Priority request from domestic hot water (BI).

Outputs

Heating circuit block output signals (hardware and logic).

Output Parameter	Description
Heating Valve Control	Control valve in the supply coil (AO)
Pump Command	Start signal for primary side pump (BO)
Pump Status	Multistate pump status point (MO)
Pump Run Alarm	Fail to stop alarm active (BO)
Pump Stop Alarm	Fail to start alarm active (BO)
Pump Fault Alarm	Direct fault alarm active (BO)
Effective Water Temp Set Point	Effective water temperature set point value (AO)

Block Functions

Heating Circuit Block Cooperation with Heat Generation Control block

The Heating Circuit block is supposed to be used as subordinate of the Heat Generation Control block. The Heating Circuit gets *EnableCommand* from Heating Curve block. The Heating Circuit block returns the Heat Demand signal to the Heat Generation Control block to activate heat generation.

Forced Cooldown Protection

This block checks if *SecSupWaterTemp* and *PrimRetWaterTemp* sensors are connected, then chooses the higher value to determine the effective supply water temperature *EffSecSupWaterTemp*.

This procedure is based on the assumption that If *PrimRetWaterTemp* is higher than *SecSupWaterTemp*, then it leads to a conclusion that all heat recipient circuit valves are closed and water in the secondary circuit is stagnant, thus it cannot give reliable information on actual supply temperature from the heat generators.

The resulting *EffSecSupWaterTemp* value must be higher than *RetWaterTemp + C.SupToCircTempDiff* in order to enable valve operation.

Heating Circuit Startup and Stopping

When the block receives the EnableCommand command it enters the startup cycle.

1) Pump is started with the delay defined by point *C.PumpStartDelay.*

2) Checks if the forced cooldown protection enables valve opening.

3) After the delay, it sets the starting point of the *EffWaterTempSp* ramp to the current value of the *MixedWaterTemp.*
4) Starts the valve regulation
5) Ramps up *EffWaterTempSp* to the value coming from the *HeatingCurve* input with all the additional factors applied (*Max/MinLimit, NightSetback*)

When EnableCommand command is switched off, the block enters the stopping cycle.

1) Regulation is stopped and valve is closed.
2) Activation of the pump switch off timer with the delay defined by *C.PumpStopDelay*.
3) When the delay time elapses, it switches off the pump.

If hydraulic setup consists of a 2-way control valve without an additional bypass, then *C.PumpStopDelay* must be set to 0s to switch off the pump immediately after valve operation is disabled.

NightSetback Function

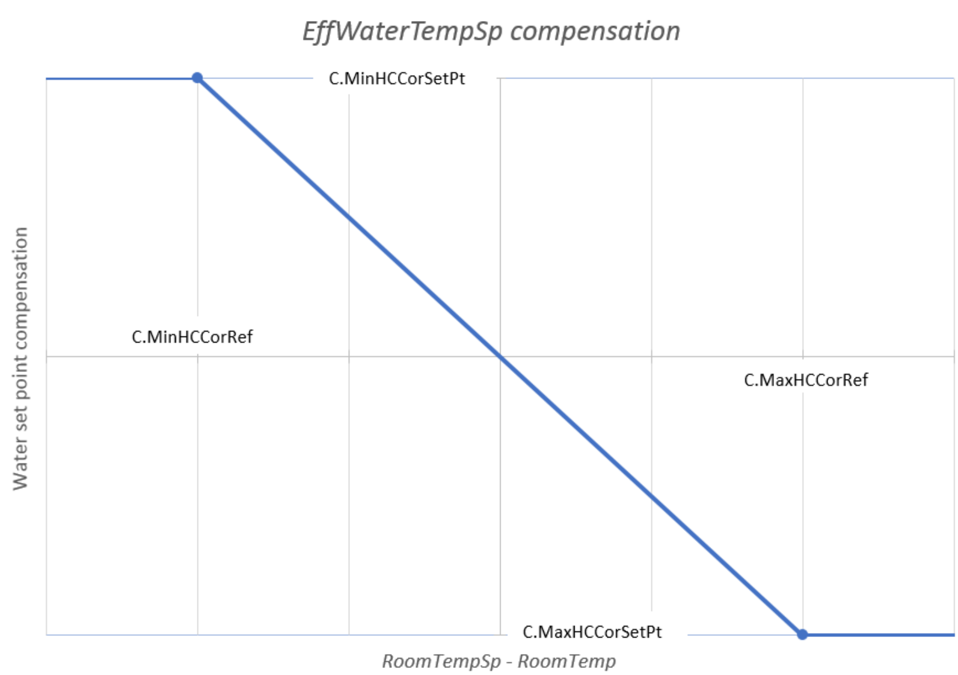
When *NightSetbackEnable* point is active, the block will lower the main control setpoint value by *NightSetbackOffset*. This limits power consumption without causing a full cooldown by switching the system into inactive mode.

RoomTemp based compensation of the HeatingCurve

RoomTemp Based Compensation of the *HeatingCurve*

If the *RoomTemp* sensor is connected, its value will be used to create an additional compensation to the *EffWaterTempSp*.

If the *RoomTemp* is below the *RoomTempSp* by *C.MinHCCorRef,* the correction will be equal to the *C.MaxHCCorSetPt*. When the *RoomTemp* is above the *RoomTempSp* by *C.MaxHCCorRef* the correction will be equal to the *C.MinHCCorSetPt*. In between these two points, the correction will vary linearly as shown on the figure below. The resulting correction value will be applied to the *EffWaterTempSp* with use of the first order filter with the time constant defined by *C.MinHCCorTConst*.



Low Room Temperature Protection

If a *RoomTemp* sensor is present and its reading drops below *C.LowRoomTempSetPt*, the heating coil will be immediately switched on and *EffWaterTempSp* will be set to *MaxLimit* value.

When *RoomTemp* rises by 2 degrees over the *C.LowRoomTempSetPt* value, the system returns to normal operation.

MixedWaterTemp Control

This block uses the *HeatingCurve* input. It should be adjusted to match the type of heating coils used (additional application block is available to provide this functionality):

- High temperature radiator circuit
- Low temperature radiator circuit
- Floor heating circuit
- Fan coil heating circuit

The *HeatingCurve* value is ramped with the change ratio defined by *C.RampRatio* parameter to obtain *EffWaterTempSp*. The ramp starting point is set equal to the *MixedWaterTemp* value at the moment when temperature regulation is enabled.

PI controller compares *EffWaterTempSp* with *MixedWaterTemp* value and calculates *HeatValveCtrl* signal which is used to control the 3-way mixing valve output.

Pump Operation and Alarm

Pumps are started according to the startup procedure described in *Heating Circuit Startup and Stopping*.

Pump status signals can be connected, and depending on the *PumpStatus*, *PumpStartCmdFb* and *PumpFault* parameters, they can generate alarms in different ways such as:

- *PumpFault* is treated as a direct fault signal and the pump alarm will be generated immediately when *PumpFault* input is true.
- *PumpStartCmdFb* and *PumpStatus* are treated as pump operation feedback signals. *PumpStatus* has priority over *PumpStartCmdFb* and if both are connected, the latter is ignored. If feedback fails to follow *PumpCmd* signal, *RunAlarm* (fail to stop) and *StopAlarm* (fail to start) signals will be generated (with corresponding *C.RunAlmDelay* and *C.StopAlmDelay* times).

An algorithm outputs a *PumpStatus.* This multistate output signal provids a coded pump status, with the following values:

- 1 – Stop
- 2 – Start
- 3 – Run Alarm (fail to stop)
- 4 – Stop Alarm (fail to start)
- 5 – Fault Alarm

High Temperature Alarm

If the *HighTempThermostat*signal is activated, the valve/valves will be shut off immediately and the pump will continue to operate.

External Overrides

This block will accept the following overrides.

Override Parameter	Description
Boiler Overheat	Used for the emergency requirement to disperse heat from boilers. It forces the pump to start and sets *EffWaterTempSp* to *MaxLimit* in order to dissipate heat from the overheated boilers.
Domestic Hot Water Priority	Used if the domestic hot water system signals priority override (the state when the hot water buffer needs to be heated up and the heating generation system lacks the capacity to satisfy increased demand). The valve will be closed for the time of the override. It is not recommended to use this function on coils suppling AHU units, due to very high sensitivity of those coils to rapid temperature drop in the supply water temperature and risk of heater freezing.
Pump Exercise	Used if the system is inactive. The pump will be switched on periodically in order to prevent blocking due to limescale buildup. The Pump Exercise has two phases: 1) Valve is opened for *C.ValveStrokeTime*, then closes. 2) Pump is started for *C.PumpExerciseTime.*

Domestic Hot Water Control Block (HTG_DomesticHotWater)

Domestic Hot Water systems are necessary to provide hot tap water for hygienic use and consumption. Regarding its use, special measures must be undertaken to ensure water is of the required quality (especially in regard to legionella bacteria elimination).

The HTG_DomesticHotWater control block provides control for installations with a buffer tank. Water is taken from the main supply connection point and warmed up in the tank to the required temperature where it is then distributed throughout the building. With this approach, maximum water temperature is controlled at the level of the buffer tank.

Inputs

The Domestic Hot Water control block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Buffer Water Temperature	Temperature of water supplied to the circulation coil (AI)
Buffer Water Temperature Set Point	Temperature set point for water supplied to the circulation coil (AI)
Reset	Alarm reset command (BI)
DHW Enabled	DHW system activation (BI)

Optional Inputs

Input Parameter	Description
High Temperature Thermostat	Thermostat that is used to immediately cut off valve in case of circuit overheating (BI)
Lower Buffer Temperature sensor	Used to control water heating in big buffer tanks (AI)
Night Setback Offset	Water temperature difference by which buffer set point is decreased when night setback function is active (AI)
Night Setback Enable	Point activating night setback function (BI)
Supply/Circulation Pump Status	Confirms operation of the pump (feedback from a contact or a flow switch).Do not use simultaneously with Pump Start Command Feedback. (BI)
Supply/Circulation Pump Start Command Feedback	Confirmation signal from the output of the controller that it is energized. Do not use simultaneously with Pump Start Command Feedback. (BI)
Supply/Circulation Pump Fault	Fault signal of the pump (feedback from a contactor or a flow switch) (BI)
Secondary Supply Water Temperature	Value of the water temperature at the supply of the main heating coil (AI)
Primary Return Water Temperature	Value of the water temperature at the return to heat generators (AI)
Activate Legionella Protection	Activation command for the anti-legionella procedure (BI)
Boiler Overheat	Heat dissipation request from overheated boiler system (BI)
Pump Exercise	Activation command for periodic pump and valve motorisation (BI)

Outputs

Domestic hot water output signals (hardware and logic).

Output Parameter	Description
Heating Valve Control	Control valve in the supply coil (AO)
Supply Pump Command	Start signal for primary side pump (BO)
Circulation Pump Command	Start signal for secondary side circulation pump (BO)
Supply/Circulation Pump Status	Multistate pump status point (MO)
Supply/Circulation Pump Run Alarm	Fail to stop alarm active (BO)
Supply/Circulation Pump Stop Alarm	Fail to start alarm active (BO)
Supply/Circulation Pump Fault Alarm	Direct fault alarm active (BO)
Effective Buffer Water Temp Set Point	Effective water temperature set point value (AO)
Legionella Alarm	Legionella protection failed alarm (BO)
DHW Priority	Domestic hot water priority request (BO)

Block Functions

System Configuration

Domestic Hot Water systems can have different hardware configurations.

Hardware Configuration	Description
1 – 3-way primary valve	Mixing valve is located on the return pipe. This allows the starting of the pump before opening of the valve. *PrimWaterTemp* sensor may be located in the coil behind a pump.
2 – 2-way primary valve	Pump must be started after opening of the valve. *PrimWaterTemp* sensor must be placed in the supply manifold.
3 – No primary valve	On/off control executed by pump switching. *PrimWaterTemp* sensor must be placed in the main coil.

Forced Cooldown Protection

This block checks if *SecSupWaterTemp* and *PrimRetWaterTemp* sensors are connected. It then chooses the higher of the two values to determine the effective supply water temperature (*EffSecSupWaterTemp*).

This procedure is based on the assumption that if *PrimRetWaterTemp* is higher than *SecSupWaterTemp*, it concludes that all heat recipient circuit valves are closed and water in the secondary circuit is stagnant, thus it cannot give reliable information on actual supply temperature from the heat generators.

The resulting *EffSecSupWaterTemp* value must be higher than *BufferWaterTemp* + *C.SupToCircTempDiff* in order to enable correct valve operation.

Startup and Shutdown

This system is enabled by *DHWEnabled.* It causes the circulation pump to start and initializes buffer tank temperature control.

If *BufferWaterTemp* is below its setpoint, heating up of the buffer will be started differently based on the valve type being used.

3-Way Valve:

1) Pump is started.
2) If *EffSecSupWaterTemp* value is present, the valve is opened when sensor measurement is higher than *BufferWaterTemp* by no less than *C.SupToCircTempDiff* parameter (default 10°C).
3) Buffer temperature control will be activated.

2-Way Valve or No Valve:

1) If *EffSecSupWaterTemp* value is present, the valve is opened when sensor measurement is higher than *BufferWaterTemp* by no less than *PrimBufferTempDifference* parameter (default 10°C).
2) Buffer temperature control is activated.
3) Pump is started when valve is opening reaches 10%.

NightSetback Function

When *NightSetbackEnable* point is active, the block will lower the main control setpoint value by *NightSetbackOffset*. This limits power consumption without causing a full cooldown by switching the system into inactive mode.

BufferWaterTemp Continuous Control

If *LowerBufferTemp* sensor is connected, then control range will be divided into two sections:

1) *BufferWaterTemp* sensor will generate the control signal based on P controller, with bias set to 50% and setpoint equal to *DomHotWaterSetpt* + (PID_{Pband}/2)
2) *LowerBufferTemp* sensor will generate the control signal using the PD controller with its set point adjusted in accordance with the same rule as for the *BufferWaterTemp* control.

If *LowerBufferTemp* sensor is **not** connected, the buffer temperature will be controlled by PI controller with bias set to 50% so that the control action starts before the temperature drops below the set point.

BufferWaterTemp on/off Control

If *LowerBufferTemp* sensor is not connected, *PrimaryPump* will be turned on when *BufferWaterTemp* temperature drops below *EffBufferWaterTempSp* minus *C.PBandTankCtrl* and will be switched off only when the temperature rises above the set point.

If *LowerBufferTemp* sensor is connected, *PrimaryPump* will be switched on when *BufferWaterTemp* temperature drops below the setpoint, and will be switched off when both *BufferWaterTemp* and *LowerBufferTemp* will show values higher than the setpoint.

High Temperature Alarm

If *HighTempThermostat* signal is activated, the *valve will be closed immediately and the circulation pump will be stopped.* It is important to set this thermostat to a higher value than the legionella protection setpoint.

Legionella Protection Function

The rising edge at the *ActLegProt* input starts the legionella protection function. *C.LegionellaProtSetPt* sets the desired value of the buffer tank in the legionella protection mode. This value should not be less than 65°C. If this value is 0, then the function is disabled.

BufferWaterTemp and, if connected, *LowBufferWaterTemp* must rise above *C.LegSetPtDecreaseRatio*. When the set point has been reached, it will be sustained for a period defined by *C.LegFuncAktTime* after which the function will be considered successfully terminated and buffer temperature setpoint will be ramped down to its normal value (ramp ratio is defined by *C.LegSetPtDecreaseRatio*).

If *C.LegionellaProtSetPt* could not be reached during time *C.TimeToLegSetpt,* an alarm will be generated.

Pump Operation and Alarms

Pumps are started according to the startup procedure described in Startup and Shutdown.

Pump status signals can be connected, and depending on the *Sup/CircPumpStatus*, *Sup/CircPumpStartCmdFb* and *Sup/CircPumpFault* parameters, they can generate alarms in different ways such as:

- *PumpFault* is treated as direct fault signal and the pump alarm will be generated immediately when *PumpFault* input is true.
- *PumpStartCmdFb* and *PumpStatus* are treated as pump operation feedback signals. *PumpStatus* has priority over *PumpStartCmdFb* and if both are connected, the latter is ignored. If feedback fails to follow *PumpCmd* signal, *RunAlarm* (fail to stop) and *StopAlarm* (fail to start) signals will be generated (with corresponding *C.RunAlmDelay* and *C.StopAlmDelay* times).

Algorithm outputs a *Sup/CircPumpStatus* . This multistate output signal provids coded pump status, with following values:

- 1 – Stop
- 2 – Start
- 3 – Run Alarm (fail to stop)
- 4 – Stop Alarm (fail to start)
- 5 – Fault Alarm

If a secondary supply pump alarm is activated, regulation is stopped immediately and the valve is shut off.

DHW Priority

If DHW valve is open at 100% for a time longer than *C.DHWPrioStartDelay* and its function is active *(C.DHWPrioEnable* = TRUE), then *DHWPriority* request is activated. It causes all the heating consumer circuit blocks, which receive the request, to decrease their setpoints to minimum values in order to transfer available heat to the DHW system.

The maximum duration of this override is defined by the *C.DHWPrioMaxAtcTime* parameter.

General Function Blocks

General function blocks are used as subordinate blocks in other components. They provide standardized solution for multiple, commonly used tasks.

Pump Excercise

If the system is inactive, the pump will be switched on periodically in order to prevent blocking due to limescale build up.

Pump Exercise has two phases:

- The valve is opened for 120s, then it is closed
- The pump is started for 120s

Inputs

The Pump Excercise Block requires connection of input signals (hardware and logic).

Required Inputs

Input Parameter	Description
Control Active	Normal operation of the corresponding component. (DI)
Enable	Activation command for the Pump Exercise. (DI)
Valve Override Active	Valve Exercise function is active. (DO)
Valve Override Value	Valve position override. (AO)
Pump Override Active	Pump Exercise function is active. (DO)

Optional Inputs

Input Parameter	Description
Valve Running Time	Time of the valve transition from 0%-100%. (AI)
Pump Running Time	Time of the exercise pump operation. (AI)
Activation Delay	Used to avoid simultaneous activation of all the pumps and valves. (AI)
Min Inact Period	Minimum time for which *ControlActive* must be FALSE in order for the Pump Exercise to be activated.

When *Enable* input is activated and *ControlActive* is not active for at least Min Inact Period, then after a time defined by *ActivationDelay,* the valve is overridden to 100% for 120% of the ValveRunningTime. The valve is then overridden to 0% for the same period of time. Next, *PumpOverrideActive* is set to TRUE for the period of *PumpRunningTime*.

If at any moment of the pump/valve exercise action Control Active becomes TRUE, the function is immediately cancelled and overrides are released.