

Automation system for setpoint temperatures based on adaptive comfort: an in-depth guide of ACCIS capabilities running with EnergyPlus.

Daniel Sánchez-García^{1*}, Jorge Martínez-Crespo¹, Ulpiano Ruiz-Rivas Hernando¹ and Carmen Alonso²

¹ Appropriate Technology for Sustainable Development Group, University Carlos III of Madrid, Spain;

* email: dsgarcia@ing.uc3m.es;

² Department of construction. Eduardo Torroja Institute for Construction Science. CSIC.

Abstract: The possibility for energy savings from implementing setpoint temperatures based on adaptive comfort has recently been investigated. This can be achieved by using the recently proposed Adaptive-Comfort-Control-Implementation Script (ACCIS), that is a computational approach which extends the air-conditioning usage to adaptive comfort. According to both an Input Data File (IDF) and the setup supplied by the user, ACCIS turns PMV-based into adaptive setpoint building EnergyPlus models. Originally, ACCIS was an EnergyManagementSystem (EMS) script, but its capabilities have been expanded, and it has been nested in the Adaptive Comfort Control Implemented Model ("accim") Python package. This study focuses on the development of ACCIS, and provides a deeper view of the operation of the computational approach based on the relationship between the objects that compose it. As a result, ACCIS brings up new possibilities for conducting energy simulations on a global scale rather than merely on a national or continental size, since all weather file quantity limits have been tackled and now studies can be developed based on millions of simulations. Also, ACCIS could be applied to energy management, or coupled with artificial intelligence to develop smart thermostats.

Keywords: Adaptive thermal comfort, Energy efficiency, Heating and cooling setpoint temperatures, Computational approach

1. Introduction

Climate change has been a major issue for years. Climate change is worsening year after year as a result of human actions made without regard for the environment (World Wildlife Fund, 2014). Temperature rises might range from 2.5°C in 2050 to 4.5°C in 2100, 7.5°C in 2200, and 8°C in 2300 (Meinshausen *et al.*, 2011), using the preindustrial mean surface temperature as a baseline. The building industry is mandated to cut greenhouse gas emissions by 90% by 2050 (European Commission, 2011) in order to limit both pollutant gas emissions and resource depletion.

Therefore, the main purpose of buildings, which is offering shelter from the environment, is crucial in a climate change context. The human is capable to adapt physiologically and psychologically to changes in weather to a certain extent, and thus is expected to adapt the increasing temperatures. There are currently 2 main approaches to thermal comfort: adaptive thermal comfort and Predicted Mean Vote (PMV) index. Adaptive thermal comfort has been studied by means of field work mainly in buildings without active heating or cooling systems alternatively to the PMV index, which was developed by Fanger in climate chambers inspired on air-conditioned offices, where occupants were not free to take actions to adapt to the indoor environment, such as choosing the clothes they wore. Therefore, given the present

and future scenarios, the research of adaptive thermal comfort is of great importance, since it leads to the Integration of users' climate adaptability into the low energy consumption.

To keep the indoor temperature within acceptable levels, air conditioning systems are sometimes utilized. However, using these systems, which often have highly limited setpoint temperatures that are quite pleasant for the human body, may reduce people's physiological, psychological, and behavioral capacity to adjust to ambient temperature fluctuations (Humphreys, 1996), in addition to consuming a significant amount of energy. Many researches have recently been focused on reducing this problem by expanding the use of air conditioning to adaptive comfort, that is, by adopting adaptive setpoint temperatures (AST). These ASTs are setpoint temperatures calculated using adaptive thermal comfort methods, and their values correspond to upper and lower comfort limits. Strictly, the use of adaptive comfort models is only applicable when no active heating and cooling systems are used. However, a recent study provided evidences that people apparently adapt to a much broader range of indoor temperatures, regardless of the technology delivering the indoor climate (Parkinson *et al.*, 2020). As a result, because the use of AST indicates that the operating temperature fluctuates within the adaptive thermal comfort zone, it is envisaged that energy consumption will be reduced due to the use of less-restricted setpoint temperatures while maintaining adequate thermal human thermal sensation: (i) R.P. Kramer *et al.* (Kramer *et al.*, 2015) investigated energy conservation at an Amsterdam museum by employing a variety of setpoint temperatures and adaptive thermal comfort levels. The lower comfort limit was allocated to the heating setpoint temperature, and HVAC systems were only operated during museum opening hours, resulting in a 74 percent energy savings; (ii) Sánchez-Guevara Sánchez *et al.* (Sánchez-Guevara Sánchez *et al.*, 2017) used setpoint temperatures in three residential structures in Spain, based on the comfort limitations of the simplified model from the ASHRAE Standard 55-2013 (ANSI/ASHRAE, 2013), which vary monthly. In both heating and cooling, energy consumption was lowered by 20% and 80%, respectively. As a result, adaptive comfort procedures are important for the inhabitants of all buildings, including air-conditioned ones. Table 1 summarizes some of the research publications that are important to this computational technique. Given the significant energy savings potential of this energy conservation technique, it has the potential to assist many buildings, particularly offices (Gangoellis *et al.*, 2020), in reducing their energy usage (Gangoellis *et al.*, 2019).

Table 1. Relevant research articles related to ACCIS.

Authors	Citation	Setpoints	Energy Saving (%)	Level of automation
R.P. Kramer et al.	(Kramer <i>et al.</i> , 2015)	Adaptive setpoints (based on Adaptive Thermal Guideline)	74	Not specified
Sánchez-Guevara Sánchez et al.	(Sánchez-Guevara Sánchez <i>et al.</i> , 2017)	Adaptive setpoints (ASH-RAE Standard 55-2013)	80 (Cooling); 20 (Heating)	Low (based on several monthly simulations and following merge)
Sánchez-García et al.	(Sánchez-García, Bienvenido-Huertas, <i>et al.</i> , 2019)	Adaptive setpoints (EN 15251:2007)	between 23 and 46	Medium (based on Schedule:Compact objects)
Sánchez-García et al.	(Sánchez-García <i>et al.</i> , 2020)	Adaptive setpoints (EN 15251:2007)	between 31 and 70	Medium (based on Schedule:Compact objects)
Sánchez-García et al.	(Sánchez-García, Rubio-Bellido, <i>et al.</i> , 2019)	Adaptive setpoints (EN 15251:2007)	between 40 and 62	Medium (based on Schedule:Compact objects)
Sánchez-García et al.	(Sánchez-García <i>et al.</i> , 2021)	Adaptive setpoints (EN 16798-1:2021)	82.8	High (by means of accim)

Building energy simulations (BES) were primarily used to develop preliminary studies with DesignBuilder (DesignBuilder Software Ltd, 2021), which employs the simulation engine EnergyPlus (Crawley *et al.*, 2001). Furthermore, mainly 2 methods were used to determine the setpoint temperatures before the development of accim: on the one hand, very basic methods like changing setpoint temperatures monthly and then summing the results of the 12 simulations, and on the other hand, more effective methods like using Schedule:Compact objects, which are EnergyPlus objects that can define setpoint temperatures and operation schedules in detail. However, a lengthy, difficult, and time-consuming process was carried out manually, making it error-prone. Setpoint temperatures, for example, were previously computed with an Excel spreadsheet, and then copied and pasted into Schedule:Compact objects, the relevant EPW file was then allocated, and finally the combination of setpoint temperature and climatic zone was simulated. Other file management activities were also necessary.

The ACCIS is a computational technique for extending air-conditioning usage to adaptive thermal comfort (Sánchez-García *et al.*, 2021). ACCIS was developed within the scope of various research studies, known as the Adaptive-Comfort-Control-Implemented Model (ACCIM). Some parameters, which make up the input arguments for Comfort Mode (Sánchez-García, Bienvenido-Huertas, *et al.*, 2019), Category (Sánchez-García *et al.*, 2020), and HVAC mode (Sánchez-García, Rubio-Bellido, *et al.*, 2019), have been investigated. Furthermore, prior versions of ACCIS were employed, allowing for investigations based on a large number of simulations in various locations and ASTs, such as 780 (Bienvenido-Huertas *et al.*, 2020) or 48,786 (Bienvenido-Huertas *et al.*, 2021) distinct location-AST combinations. The program or process described in this paper is currently the only one available. Other approaches get

similar outcomes, but they are manual and hence time-consuming, tiresome, and error-prone.

The Python package `accim` and the EMS script `ACCIS` have already been studied in a previous research (Sánchez-García *et al.*, 2021). However, that research provided a general definition of both, but did not explain the details of these. This study provides an in-depth insight into `ACCIS`, considering the use of the different objects in the `EnergyPlus` and `EnergyManagementSystem` environments, their role and function, as well as the operation and relationship between the different variables. Section 2 provides the previous considerations, including a brief presentation of the software on which `ACCIS` depends, the different objects used in the `EnergyManagementSystem` environment, and the limitations of `ACCIS`. Section 3 provides of the operation of `ACCIS`, broken down into the different objects that constitute it. Section 4 provides the discussion, and lastly Section 5 provides the conclusions.

2. Previous considerations

2.1. `EnergyPlus` and `EnergyManagementSystem`

`EnergyPlus`, the `ACCIS` simulation engine, is an open-source BES program developed by the US Department of Energy. Engineers, architects, and academics use it to simulate energy and water use in structures. This application is a console-based program that receives data from the user and writes it to text files. It was initially created in the FORTRAN programming language, but with version 8.2.0, it has been rewritten in C++. Because this software and its programming language are hard to understand, graphical user interfaces such as `DesignBuilder` or `OpenStudio` [33] have been created to help those who have no background in programming.

`EnergyPlus` is a strong software with few customization options. Nonetheless, there is an integrated component called `EnergyManagementSystem` (EMS) that compensates for this flaw. This is a strong tool for creating bespoke control and modelling routines as well as providing supervisory control over specific areas of `EnergyPlus` modelling. It is, however, difficult to utilize since it necessitates the creation of computer programs in the `EnergyPlus` Runtime Language (ERL), which is used to explain the control algorithms. When the simulation is conducted, `EnergyPlus` understands and runs the ERL program.

2.2. `EnergyManagementSystem` objects

In order to fully understand the operation of `ACCIS`, some `EnergyManagementSystem` typical objects need to be explained: *`EnergyManagementSystem:Sensor`* (hereinafter, *sensor*) objects are used to collect data from elements of the model that might change over time inside a single run period for use in control calculations. *`EnergyManagementSystem:Actuator`* (hereinafter, *actuator*) objects are used to actuate selected features inside `EnergyPlus`. Rather of adding a new set of EMS-aware controls and component models, they usually override existing functionalities. *`EnergyManagementSystem:Program`* are the objects which contain the overridden behaviour applied to the simulation run, by means of different statements, such as `set`-statements, to assign variables, `if`-blocks, to change behaviour based on conditions, or `while`-blocks, to iterate as long as some condition is met. *`EnergyManagementSystem:ProgramCallingManager`* (hereinafter, *program calling manager*) objects are used to describe the timing for when the ERL programs are run, at some specific EMS Calling Points. These EMS Calling Points are locations inside the `EnergyPlus` program where and when the EMS can be summoned to assist.

EnergyManagementSystem:GlobalVariable (hereinafter, global variable) objects are used to declare variables with a user-defined name and global scope. Global variables can be used to store intermediate results that span across ERL programs. *EnergyManagementSystem:OutputVariable* (hereinafter, output variable) objects are used to create a custom output variable that is mapped to an EMS variable. The custom output variable can then be reported to the output file using the standard EnergyPlus output mechanisms such as with the *Output:Variable* object.

2.3. Limitations

In order to simplify the script, the building for what the script has been prepared has only one zone (named *Block1:Zone1*) and one window (named *Block1_Zone1_Wall_4_0_0_0_0_0_Win*). Therefore, for buildings with multiple zones and/or windows, the EMS sensors, actuators, program calling managers, programs, output variables and global variables related to any specific zone or window must be added for all of them. All these objects include the name of the room or window in its name.

Also, so that mixed-mode can be applied, the HVAC system indoor units must be monitored. For this purpose, a VRF system for each zone must be separately modelled, where the Outdoor units are named following the pattern “VRF Outdoor Unit_BlockX_ZoneY”. The code to consider other HVAC systems is under development, but it will be included in future releases.

Further to the above, ACCIS requires the following points for the correct operation:

- “People” objects include “AdaptiveASH55” and “AdaptiveCEN15251” in the “Thermal Comfort Model 1 Type and 2 Type” fields, otherwise the PMOT and RMOT would not be computed by EnergyPlus.
- “Schedule:Compact” objects following the name pattern “FORSCRIPT_AHST_BLOCKX_ZONEY” must exist in the IDF file, and these must be assigned to the related “BLOCKX_ZONEY” by means of the corresponding “ThermostatSetpoint:DualSetpoint” object.
- So that mixed-mode can operate, the ventilation method must be “Calculated”, natural ventilation must be available for all zones, and windows must be open always. ACCIS will close the windows if environmental conditions are not favourable.

Therefore, to apply setpoint temperatures by either copying and pasting ACCIS in the IDF or in the DesignBuilder Scripting module, these conditions must be met. Thus, the recommended and easiest method is to use accim, since it automatically checks if the IDF file comply with all limitations previously stated, and if not, automatically amends the IDF so that the operation is correct.

3. Script operation

The ACCIS code is included in the Appendix A, and it is available in open-source at the Python Package Index. It has been published in (Sánchez-García *et al.*, 2021), and a how-to-use guide can be consulted at the documentation (Sánchez-García, 2021). ACCIS can be broken down into 4 main stages: (i) Stage 1, when some user-defined data is stored at some global variables, and sensors, actuators and global variables are added; (ii) Stage 2, when some programs are executed to apply the user-defined input data to set up the customised outputs; (iii) Stage 3, when data to be reported is computed and actuators override the values of the components; and (iv) Stage 4, when the output variables are added to indicate which variables

need to be reported. The operation at all stages is explained below, and it is recommended to follow the script included in Appendix A as this section is read as well as the flowcharts (Figures 1 and 2). Comments and clarification in the code are marked with “!” sign, in grey color; highlighted syntax has been preserved.

3.1. Stage 1

(i) Firstly, a ProgramCallingManager and Program objects are added, named SetInputData. As the name states, its function is to store the input data by means of set-statements to customise the simulation run afterwards. The input data is stored in the following variables: *AdapStand*, *CAT*, *ComfMod*, *HVACmode*, *VentCtrl*, *VSToffset*, *MinOToffset*, *MaxWindSpeed*, *ACSTtol*, and *AHSTtol*. These different parameters are explained in detail in the documentation website (Sánchez-García, 2021). Some input data variables treat the input values as categorical data, used in later in conditional statements. These are:

- *AdapStand* (if 0 is stored, then all computations will be performed later considering the Spanish Building Technical Code; if 1, EN16798-1 adaptive model; if 2, ASHRAE 55 adaptive model; if 3, a Japanese local adaptive model, hereinafter JPN);
- *CAT* (if 1, 2 or 3 is stored, then EN 16798-1’s Category I, II or III will be applied; if 80 or 90, then ASHRAE 55 80% or 90% acceptability levels)
- *ComfMod* (if 0 is stored, then the setpoint temperatures will be based on the static model, based on the previous input data; if 1, 2 or 3, adaptive comfort model previously inputted will be applied when RMOT or PMOT is within applicability limits; the remaining time, static setpoint temperatures are applied).
- *HVACmode* (if 0 is stored, then the full air-conditioning mode is applied and natural ventilation is not allowed; if 1, free-running mode is applied, and natural ventilation is allowed; if 2, mixed-mode is applied, therefore natural ventilation is prioritised, otherwise windows are closed and HVAC system starts to work).
- *VentCtrl* (if 0 is stored, then the ventilation setpoint temperature is assigned to the comfort temperature; if 1, it is assigned to the upper comfort limit).

All other input data variables are treated as numbers. This program was initially located at the beginning of the script to separate it from the non-editable code, since before accim was developed, the method to apply this script was by adding it to the IDF file, and therefore the user had to customise the simulation by editing these values.

(ii) A global variable object is added to declare all the different variables that are used in ACCIS. Some of these are related to some specific zone, while others are not. Further details for each global variable can be found in the Appendix.

(iii) Then, a number of sensors, which will be used to get data from certain components, are added (Table 2). Therefore, these will allow to know the values at each timestep of the Running Mean Outdoor Temperature (RMOT), Prevailing Mean Outdoor Temperature (PMOT), and for both each thermal zone and window, the zone operative temperature, the cooling rate, the heating rate, the outdoor air wind speed and the outdoor air dry-bulb temperature. These last five components, in case of the zones, refer to the zones itself or the VRF indoor units located in these zones, and in case of the windows, these refer to the zones where the windows are located, or the VRF indoor units of the zones where the windows are located.

Table 2. List of sensors.

Name	Output:Variable or Output:Meter Index Key Name	Output:Variable or Output:Meter Name
RMOT	People Block1:Zone1	Zone Thermal Comfort CEN 15251 Adaptive Model Running Average Outdoor Air Temperature
PMOT	People Block1:Zone1	Zone Thermal Comfort ASHRAE 55 Adaptive Model Running Average Outdoor Air Temperature
Block1_Zone1_OpT	Block1:Zone1	Zone Operative Temperature
Block1_Zone1_CoolCoil	Block1:Zone1 VRF Indoor Unit DX Cooling Coil	Cooling Coil Total Cooling Rate
Block1_Zone1_HeatCoil	Block1:Zone1 VRF Indoor Unit DX Heating Coil	Heating Coil Heating Rate
Block1_Zone1_WindSpeed	Block1:Zone1	Zone Outdoor Air Wind Speed
Block1_Zone1_OutT	Block1:Zone1	Zone Outdoor Air Drybulb Temperature
Block1_Zone1_Wall_4_0_0_0_0_0_Win_OpT	Block1:Zone1	Zone Operative Temperature
Block1_Zone1_Wall_4_0_0_0_0_0_Win_CoolCoil	Block1:Zone1 VRF Indoor Unit DX Cooling Coil	Cooling Coil Total Cooling Rate
Block1_Zone1_Wall_4_0_0_0_0_0_Win_HeatCoil	Block1:Zone1 VRF Indoor Unit DX Heating Coil	Heating Coil Heating Rate
Block1_Zone1_Wall_4_0_0_0_0_0_Win_WindSpeed	Block1:Zone1	Zone Outdoor Air Wind Speed
Block1_Zone1_Wall_4_0_0_0_0_0_Win_OutT	Block1:Zone1	Zone Outdoor Air Drybulb Temperature
OutT	Environment	Site Outdoor Air Drybulb Temperature

(vi) Also, a number of actuators are added (Table 3). These will allow to override the values of the schedules assigned to the heating and cooling setpoint temperatures for the existing zone (*FORSCRIPT_ACST_Sch_Block1_Zone1* and *FORSCRIPT_AHST_Sch_Block1_Zone1* respectively, where ACST and AHST stands for Adaptive Cooling / Heating Setpoint Temperatures), as well as open or closed status of the existing window.

Table 3. List of actuators.

Name	Actuated Component Unique Name	Actuated Component Type	Actuated Component Control Type
FOR-SCRIPT_ACST_Sch_Block1_Zone1	FOR-SCRIPT_ACST_Block1_Zone1	Schedule:Compact	Schedule Value
FOR-SCRIPT_AHST_Sch_Block1_Zone1	FOR-SCRIPT_AHST_Block1_Zone1	Schedule:Compact	Schedule Value
Block1_Zone1_Wall_4_0_0_0_0_0_Win_VentOpenFact	Block1:Zone1_Wall_4_0_0_0_0_0_Win	AirFlow Network Window/Door Opening	Venting Opening Factor

3.2. Stage 2

Once the global variables, sensors and actuators have been added, the next objects are the programs, which will determine the relationship between the previously added sensors and actuators, and will define the different global variables declared. Also, the program calling managers are added, which will determine at which stage should the programs be called.

First, the programs used to determine the settings of the adaptive setpoint temperatures are added. These will compute and store the corresponding values to some global variables based on the input data specified at the *SetInputData* program.

SetComfTemp is used to store the comfort temperature at each timestep in the global variable *ComfTemp*. It depends on the value assigned to *AdapStand* at the *SetInputData* program: if 1, then *ComfTemp* will store the comfort temperature for EN16798-1; if 2, ASHRAE 55; if 3, JPN. In case of 1, the *RMOT* sensor is used to compute the comfort temperature, while in case of 2 or 3, *PMOT* sensor is used instead.

SetAppLimits is used to store the values of the applicability limits in the global variables added for that purpose (*AHSTaul*, *AHSTall*, *ACSTaul* and *ACSTall*). Depending on the value assigned to *AdapStand* as well, the applicability limits will be set for the corresponding comfort model. If the stored value is not any of the previously stated, then applicability limits are set to 50 (which means the model will not be applicable).

ApplyCAT is used to store the values of the offset for the heating and cooling setpoint temperatures from comfort temperature in the global variables added for that purpose (*AHSToffset* and *ACSToffset*). These also depends on the values previously stored in *AdapStand* and *CAT*.

SetAST is used to compute and store the values for the setpoint temperatures in the global variables *AHST* and *ACST*. It depends on the values previously store for *AdapStand* and *ComfMod*, as well as other sensors and global variables. This is the longest and one of the most important programs, since it consists of 25 if-blocks which contains the following set-statements patterns to compute and store the value for the setpoints:

$$\text{set ACST} = \text{RMOT} * m + n + \text{ACSToffset} + \text{ACSTtol} \quad (1)$$

$$\text{set AHST} = \text{RMOT} * m + n + \text{AHSToffset} + \text{AHSTtol} \quad (2)$$

where “set” is the command for ERL to store the value; *ACST* and *AHST* are respectively the global variables for the adaptive cooling and heating setpoint temperatures; *RMOT* is the sensor for the running mean outdoor temperature, which would be *PMOT* in case of ASHRAE 55 or JPN; “m” and “n” are coefficients of the comfort model regression equation (for example, 0.33+18.8 in case of EN 16798-1 adaptive model); *ACSToffset* and *AHSToffset* are global variables for the offset from comfort temperature previously stored in *ApplyCAT*; and *ACSTtol* and *AHSTtol* are global variables added to allow for certain tolerances, stored in *SetInputData*. Since sometimes, the simulated operative temperature might be some decimals of degree exceeding the setpoint temperatures, these tolerances, whose typical values could be $\pm 0.1^\circ\text{C}$, are meant to be used to ensure that temperature in all timesteps falls within the comfort zone.

SetASTnoTol is used to compute and store the setpoint temperatures without tolerances (*AHSTnoTol* and *ACSTnoTol*), which are, indeed, the same value as the adaptive comfort limits. The value stored at *ACSTnoTol* must be negative, while the value stored at *AHSTnoTol* must be positive.

SetVST is used to compute and store the minimum outdoor temperature to apply the mixed-mode and the ventilation setpoint temperature at global variables *MinOutTemp* and *VST*. The value for *MinOutTemp* is computed by subtracting the global variable *MinOToffset* (stored at *SetInputData*) to the *AHST*, while *VST* is computed by adding the global variable *VSToffset*

(also stored at *SetInputData*), when the adaptive model is applicable; otherwise, it is stored as the mean of both setpoint temperatures.

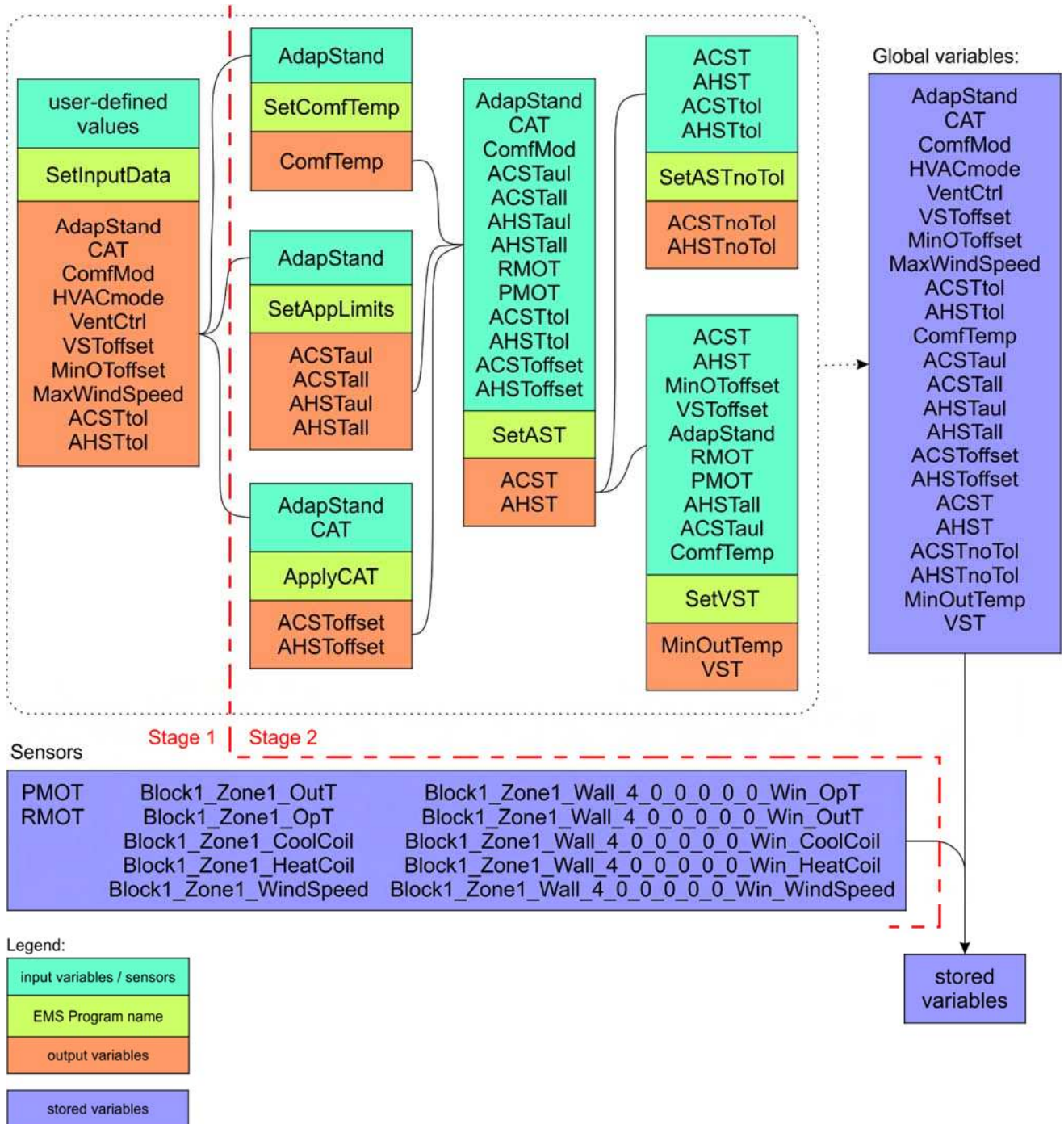


Figure 1. Stages 1 and 2 of ACCIS.

3.3. Stage 3

Up to now, all programs have been used to determine the settings of the setpoint temperatures. Therefore, all the following programs are used to perform calculations or override the setpoint temperatures schedule object and the open-or-close state of the window.

CountHours_Block1_Zone1 is used to compute the time when the related zone's operative temperature (i) is within thermal comfort limits, when it is not because of heat (ii) or cold (iii)

although still being applicable, and when it is not applicable because of heat (vi) or cold (v) and finally store it in the global variables (i) *ComfHours_Block1_Zone1*, (ii) *DiscomfAppHotHours_Block1_Zone1*, (iii) *DiscomfAppColdHours_Block1_Zone1*, (vi) *DiscomfNonAppHotHours_Block1_Zone1* and (v) *DiscomfNonAppColdHours_Block1_Zone1* respectively added for that purpose. At this point, all other variables are either sensors or global variables which has been previously stored.

CountHoursNoApp_Block1_Zone1 is similar to the previous program, however this does not consider the applicability limits, that is, only evaluates if operative temperature is within comfort limits (stored at the global variables *ACSTnoTol* and *AHSTnoTol*), and if it is, stores the time at the global variable *ComfHoursNoApp_Block1_Zone1*.

ApplyAST_Block1_Zone1 is composed by 3 if-blocks with several if-blocks nested. The first one is used to evaluate if environmental conditions are acceptable to ventilate in mixed-mode, and if so, 1 is stored as a boolean at the local variable *Ventilates_HVACmode2_Block1_Zone1* for later use at the third if-block. The second if-block is used to evaluate also if environmental conditions are acceptable to ventilate, but in free-running mode, and it depends on the global variable *VentCtrl* stored at *SetInputData*. If conditions are acceptable, 1 is stored as well as a boolean, but in a different local variable named *Ventilates_HVACmode1_Block1_Zone1*. Lastly, the third if-block overrides the values of the schedules named *FORSCRIPT_ACST_Block1_Zone1* and *FORSCRIPT_AHST_Block1_Zone1* depending on the value of the global variable *HVACmode* stored at *SetInputData*. If *HVACmode* is 0, the value of *ACST* and *AHST* is respectively assigned to the actuators *FORSCRIPT_ACST_Sch_Block1_Zone1* and *FORSCRIPT_AHST_Sch_Block1_Zone1*; if *HVACmode* is 1, 100 and -100 are respectively assigned to the cooling and heating setpoint actuators, so that EnergyPlus understands the HVAC system will not be allowed to work; finally, if *HVACmode* is 2 and the environmental conditions previously evaluated are not acceptable, *ACST* and *AHST* are assigned as well to the cooling and heating actuators. Also, the time when ventilation is allowed is stored at the global variable *VentHours_Block1_Zone1*. The local variable *VentHours_Block1_Zone1* has not been declared since its scope is limited to this program. This program works at zone level, therefore all sensors are related to thermal zones (all sensors include the zone name as a prefix or suffix).

SetWindowOperation_Block1_Zone1_Wall_4_0_0_0_0_0_0_Win is very similar to the previous program *ApplyAST_Block1_Zone1* in that the structure and purposes are the same, with the exception that the overridden component are the open-or-close status of the windows, controlled by the actuator *Block1_Zone1_Wall_4_0_0_0_0_0_0_Win_VentOpenFact* (1 is open, 0 is closed). In this case, this program works at window level, therefore all sensors are related to the window, and include the window name as a prefix or suffix.

3.4. Stage 4

Finally, the output variables for the global variables that have been declared are added. These objects are used to provide information to EnergyPlus of the EMS-related global variables that need to be reported at the simulation results. Table 4 shows the full list of output variables objects, where field “EMS Variable Name” is the name of the global variable, and “Name” and “Units” are the name and units under which these are reported.

It must be noted that “Output:Variable” objects are different to “EnergyManagementSystem:OutputVariables” objects. “Output:Variable” objects are used at

a wider scope than EMS, and these indicates the variables that need to be reported, related and not related to EMS.

Table 4. List of EMS output variables.

Name	EMS Variable Name	Units
Comfort Temperature	ComfTemp	C
Adaptive Cooling Setpoint Temperature	ACST	C
Adaptive Heating Setpoint Temperature	AHST	C
Adaptive Cooling Setpoint Temperature_No Tolerance	ACSTnoTol	C
Adaptive Heating Setpoint Temperature_No Tolerance	AHSTnoTol	C
Ventilation Setpoint Temperature	VST	C
Minimum Outdoor Temperature for mixed mode ventilation	MinOutTemp	C
Comfortable Hours_No Applicability_Block1_Zone1 (summed)	ComfHoursNoApp_Block1_Zone1	h
Comfortable Hours_Block1_Zone1 (summed)	ComfHours_Block1_Zone1	h
Discomfortable Applicable Hot Hours_Block1_Zone1 (summed)	DiscomfAppHotHours_Block1_Zone1	h
Discomfortable Applicable Cold Hours_Block1_Zone1 (summed)	DiscomfAppColdHours_Block1_Zone1	h
Discomfortable Non Applicable Hot Hours_Block1_Zone1 (summed)	DiscomfNonAppHotHours_Block1_Zone1	h
Discomfortable Non Applicable Cold Hours_Block1_Zone1 (summed)	DiscomfNonAppColdHours_Block1_Zone1	h
Ventilation Hours_Block1_Zone1 (summed)	VentHours_Block1_Zone1	h

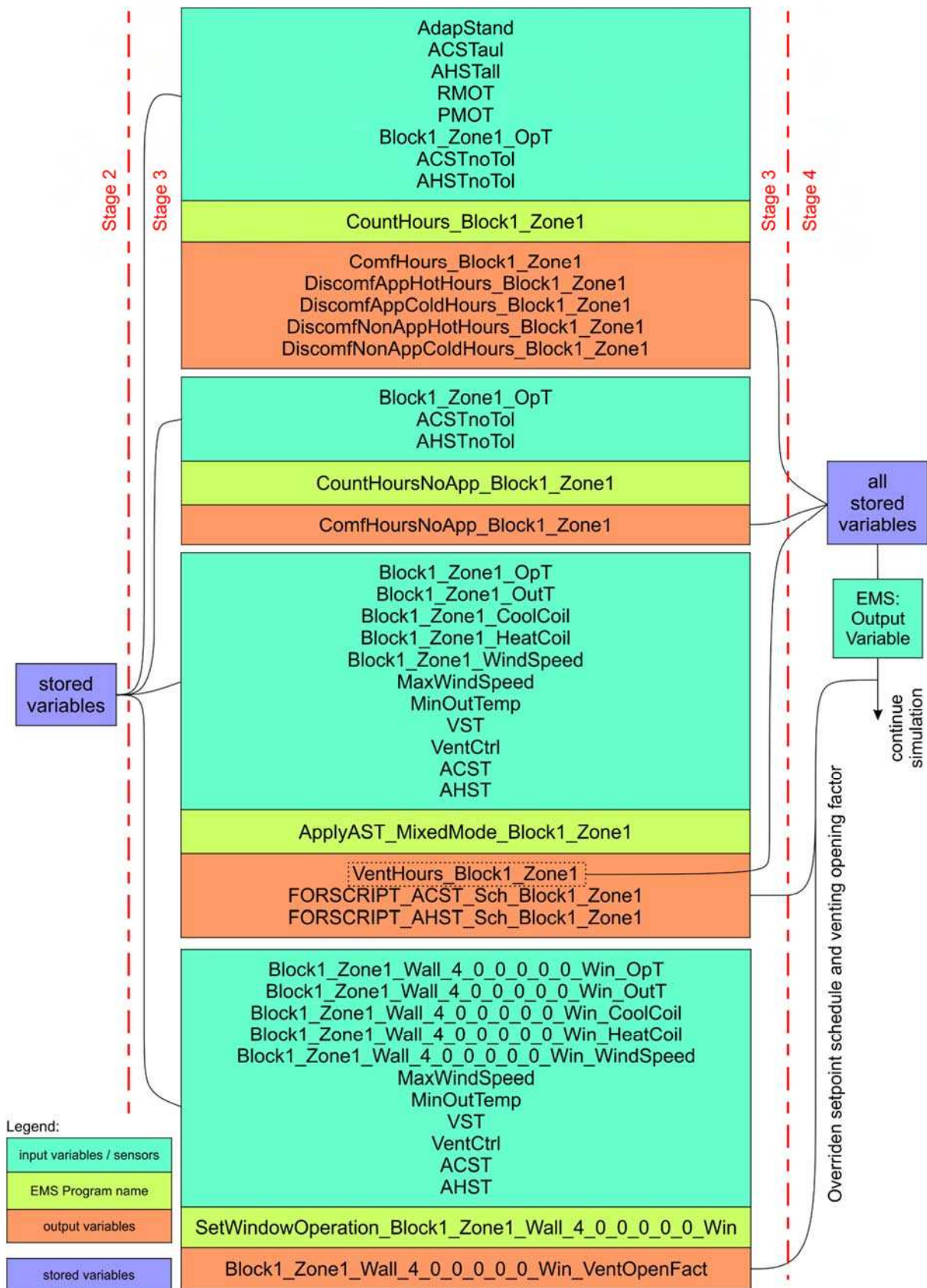


Figure 2. Stages 3 and 4 of ACCIS

4. Discussion

One of the most important features of ACCIS is that adaptive setpoint temperatures are not previously calculated for some specific climate and then pasted in the corresponding Schedule:Compact object which controls the thermostat of the thermal zones (in this code, *FORSCRIPT_ACST_Block1_Zone1* and *FORSCRIPT_AHST_Block1_Zone1*), but instead, these are calculated by EnergyPlus as the simulation runs and stored at the *RMOT* and *PMOT* sensors, and therefore this code is valid for the simulation with any climate file. Therefore, the proper operation of ACCIS relies on the availability of these sensors.

As a result, all possible setpoint temperatures are listed in Table 5, depending on the user-defined parameters *AdapStand*, *CAT* and *ComfMod*. All these setpoint temperatures make what could be understood as 'hybrid' comfort models, since when the adaptive model is applicable, adaptive setpoint temperatures are used, otherwise if it is not applicable, static setpoint temperatures are used instead (except when *ComfMod* is 0, which is entirely static).

The energy-saving potential of ACCIS, as well as the extensive simulation runs it allows, open up new avenues for performing energy simulations on a worldwide scale, rather than just on a nation or continent level. As a result, this tool opens up new opportunities for extending the scope of various adaptive comfort research investigations, such as climate change mitigation measures or energy poverty studies. Namely, this approach will be used in the Energy Poverty Intelligence Unit (EPIU) to estimate the energy consumption of different representative buildings of Getafe, in Madrid (Spain). To the best of the authors' knowledge, the suggested computational strategy is the first to do so, and it also permits additional standard adaptive-setpoint simulations to be completed in less time than any previous method. Furthermore, this technique has been automated and built for ease-of-use, whereas other systems require performing file management activities for dozens or hundreds of EPW files, and IDF files with adaptive setpoints must be simulated precisely with the relevant EPW file.

COMFORT AT THE EXTREMES 2022

Table 5. List of all available setpoint temperatures based on the different input data.

AdapStand	CAT	ComfMod	Cooling setpoint temperature (°C)			Heating setpoint temperature (°C)			
			WMOT < ACSTall	ACSTall < WMOT < ACSTaul	ACSTaul < WMOT	WMOT < AHSTall	AHSTall < WMOT < AHSTaul	AHSTaul < WMOT	
			23:00-08:00h: 27 / 08:00-23:00h: 25			23:00-08:00h: 17 / 08:00-23:00h: 20			
0	NA	0	25.5 (25 in WS)			21 (23.5 in SS)			
		1	27-25	RMOT*0.33+18.8+2	27-25	17-20	RMOT*0.33+18.8-3	17-20	
		2	25	RMOT*0.33+18.8+2	25.5	21	RMOT*0.33+18.8-3	23.5	
	1	2	3	10*0.33+18.8+2	RMOT*0.33+18.8+2	30*0.33+18.8+2	10*0.33+18.8-3	RMOT*0.33+18.8-3	30*0.33+18.8-3
			0	26 (25 in WS)			20 (23 in SS)		
			1	27-25	RMOT*0.33+18.8+3	27-25	17-20	RMOT*0.33+18.8-4	17-20
	1	3	2	25	RMOT*0.33+18.8+3	26	20	RMOT*0.33+18.8-4	23
			3	10*0.33+18.8+3	RMOT*0.33+18.8+3	30*0.33+18.8+3	10*0.33+18.8-4	RMOT*0.33+18.8-4	30*0.33+18.8-4
			0	27 (25 in WS)			18 (22 in SS)		
1	3	1	27-25	RMOT*0.33+18.8+4	27-25	17-20	RMOT*0.33+18.8-5	17-20	
		2	25	RMOT*0.33+18.8+4	27	18	RMOT*0.33+18.8-5	22	
		3	10*0.33+18.8+4	RMOT*0.33+18.8+4	30*0.33+18.8+4	10*0.33+18.8-5	RMOT*0.33+18.8-5	30*0.33+18.8-5	
2	80	0	28.35 (26.35 in WS)			20.1 (23.78 in SS)			
		1	27-25	PMOT*0.31+17.8+3.5	27-25	17-20	PMOT*0.31+17.8-3.5	17-20	
		2	26.35	PMOT*0.31+17.8+3.5	28.35	20.1	PMOT*0.31+17.8-3.5	23.78	
	2	90	3	10*0.31+17.8+3.5	PMOT*0.31+17.8+3.5	33.5*0.31+17.8+3.5	10*0.31+17.8-3.5	PMOT*0.31+17.8-3.5	33.5*0.31+17.8-3.5
			0	27.42 (25.09 in WS)			21.44 (24.74 in SS)		
			1	27-25	PMOT*0.31+17.8+2.5	27-25	17-20	PMOT*0.31+17.8-2.5	17-20
	2	90	2	25.09	PMOT*0.31+17.8+2.5	27.42	21.44	PMOT*0.31+17.8-2.5	24.74
			3	10*0.31+17.8+2.5	PMOT*0.31+17.8+2.5	33.5*0.31+17.8+2.5	10*0.31+17.8-2.5	PMOT*0.31+17.8-2.5	33.5*0.31+17.8-2.5
			0	28			18		
2	80	1	28	PMOT*0.48+14.4+3.5	28	18	PMOT*0.48+14.4-3.5	18	
		2	26.35	PMOT*0.48+14.4+3.5	28.35	20.1	PMOT*0.48+14.4-3.5	23.78	
		3	5*0.48+14.4+3.5	PMOT*0.48+14.4+3.5	30*0.48+14.4+3.5	5*0.48+14.4-3.5	PMOT*0.48+14.4-3.5	30*0.48+14.4-3.5	
2	90	0	27			19			
		1	27	PMOT*0.48+14.4+2.5	27	19	PMOT*0.48+14.4-2.5	19	
		2	25.09	PMOT*0.48+14.4+2.5	27.42	21.44	PMOT*0.48+14.4-2.5	24.74	
2	90	3	5*0.48+14.4+2.5	PMOT*0.48+14.4+2.5	30*0.48+14.4+2.5	5*0.48+14.4-2.5	PMOT*0.48+14.4-2.5	30*0.48+14.4-2.5	

PMOT: Prevailing mean outdoor temperature; RMOT: Running mean outdoor temperature; WS: Winter season; SS: Summer season; NA: Not applicable; ACSTall: Adaptive Cooling Setpoint Temperature applicability lower limit; ACSTaul: Adaptive Cooling Setpoint Temperature applicability upper limit; AHSTall: Adaptive Heating Setpoint Temperature applicability lower limit; AHSTaul: Adaptive Heating Setpoint Temperature applicability upper limit; WMOT: Weighted Mean Outdoor Temperature (can be either RMOT or PMOT)

5. Conclusions

Adaptive setpoint temperatures are a possible way to improve the energy efficiency of a building. It has been widely demonstrated that it has a significant energy-saving potential. However, because the majority of the operations were completed manually, the approach used to reach these findings was time-consuming and error-prone. A computational strategy known as ACCIS was recently presented to address this issue, and there is presently no other way capable of automating the process of applying adaptive setpoint temperatures. However, ACCIS had some limitations in that a number of requirements must be complied with so that the operation was correct. Therefore, ACCIS has been nested in a Python library called *accim*, which allows to automate all these tasks. As a result, according to an IDF and the setup supplied by the user, ACCIS turns PMV-based into adaptive setpoint building energy models.

Once stated some previous considerations, this study provides an in-depth insight of ACCIS, breaking down the code into 4 stages:

- Stage 1, where the user specifies the input data for the customisation of the adaptive setpoint simulation settings, and all needed EMS objects are added;
- Stage 2, where EMS programs to apply the user-defined input data and compute and store the global variables to define the framework of the customised adaptive setpoint simulation
- Stage 3, where the setpoint schedule and open-or-close state of the windows are overridden based on the global variables previously computed in Stage 2
- Stage 4, when the variables that are going to be reported are specified.

The energy-saving potential of ACCIS, as well as the long simulation runs it permits, bring up new possibilities for conducting energy simulations on a global scale rather than merely on a national or continental size. As a result, this instrument expands the scope of many adaptive comfort research studies, such as climate change mitigation or energy poverty study.

6. References

- ANSI/ASHRAE, 2013. *ASHRAE Standard 55-2013 Thermal Environmental Conditions for Human Occupancy*. ASHRAE Standard. GA, United States.
- Bienvenido-Huertas, D., Sánchez-García, D., Rubio-Bellido, C., and Oliveira, M.J., 2020. Influence of adaptive energy saving techniques on office buildings located in cities of the Iberian Peninsula. *Sustainable Cities and Society*, 53, 101944.
- Bienvenido-Huertas, D., Sánchez-García, D., Rubio-Bellido, C., and Pulido-Arcas, J.A., 2021. Analysing the inequitable energy framework for the implementation of nearly zero energy buildings (nZEB) in Spain. *Journal of Building Engineering*, 35 (March 2021), 102011.
- Crawley, D.B., Lawrie, L.K., Winkelmann, F.C., Buhl, W.F., Huang, Y.J., Pedersen, C.O., Strand, R.K., Liesen, R.J., Fisher, D.E., Witte, M.J., and Glazer, J., 2001. EnergyPlus: Creating a new-generation building energy simulation program. *Energy and Buildings*, 33 (4), 319–331.
- DesignBuilder Software Ltd, 2021. DesignBuilder [online]. Available from: <https://designbuilder.co.uk> [Accessed 18 Jul 2021].
- European Commission, 2011. A Roadmap for moving to a competitive low carbon economy in 2050 [online]. Available from: <https://eur-lex.europa.eu/legal->

content/EN/ALL/?uri=CELEX:52011DC0112 [Accessed 18 Jul 2021].

- Gangoellés, Casals, Ferré-Bigorra, Forcada, Macarulla, Gaspar, and Tejedor, 2019. Energy Benchmarking of Existing Office Stock in Spain: Trends and Drivers. *Sustainability*, 11 (22), 6356.
- Gangoellés, M., Casals, M., Ferré-Bigorra, J., Forcada, N., Macarulla, M., Gaspar, K., and Tejedor, B., 2020. Office representatives for cost-optimal energy retrofitting analysis: A novel approach using cluster analysis of energy performance certificate databases. *Energy and Buildings*, 206, 109557.
- Humphreys, M.A., 1996. Thermal comfort temperatures world-wide - the current position. *Renewable Energy*, 8 (1–4), 139–144.
- Kramer, R.P., Maas, M.P.E., Martens, M.H.J., van Schijndel, A.W.M., and Schellen, H.L., 2015. Energy conservation in museums using different setpoint strategies: A case study for a state-of-the-art museum using building simulations. *Applied Energy*, 158, 446–458.
- Meinshausen, M., Smith, S.J., Calvin, K., Daniel, J.S., Kainuma, M.L.T., Lamarque, J., Matsumoto, K., Montzka, S.A., Raper, S.C.B., Riahi, K., Thomson, A., Velders, G.J.M., and van Vuuren, D.P.P., 2011. The RCP greenhouse gas concentrations and their extensions from 1765 to 2300. *Climatic Change*, 109 (1), 213–241.
- Parkinson, T., de Dear, R., and Brager, G., 2020. Nudging the adaptive thermal comfort model. *Energy and Buildings*, 206, 109559.
- Sánchez-García, D., 2021. accim's documentation [online]. Available from: <https://accim.readthedocs.io/en/latest/index.html> [Accessed 16 Apr 2021].
- Sánchez-García, D., Bienvenido-Huertas, D., and Rubio-Bellido, C., 2021. Computational approach to extend the air-conditioning usage to adaptive comfort: Adaptive-Comfort-Control-Implementation Script. *Automation in Construction*, 131, 103900.
- Sánchez-García, D., Bienvenido-Huertas, D., Tristancho-Carvajal, M., and Rubio-Bellido, C., 2019. Adaptive Comfort Control Implemented Model (ACCIM) for Energy Consumption Predictions in Dwellings under Current and Future Climate Conditions: A Case Study Located in Spain. *Energies*, 12(8) (1498).
- Sánchez-García, D., Rubio-Bellido, C., del Río, J.J.M., and Pérez-Fargallo, A., 2019. Towards the quantification of energy demand and consumption through the adaptive comfort approach in mixed mode office buildings considering climate change. *Energy and Buildings*, 187, 173–185.
- Sánchez-García, D., Rubio-Bellido, C., Tristancho, M., and Marrero, M., 2020. A comparative study on energy demand through the adaptive thermal comfort approach considering climate change in office buildings of Spain. *Building Simulation*, 13 (1), 51–63.
- Sánchez-Guevara Sánchez, C., Mavrogianni, A., and Neila González, F.J., 2017. On the minimal thermal habitability conditions in low income dwellings in Spain for a new definition of fuel poverty. *Building and Environment*, 114, 344–356.
- World Wildlife Fund, 2014. *Living Planet Report 2014: Species and spaces, people and places*. Gland, Switzerland: WWF International.

IAPPENDIX A. ADAPTIVE-COMFORT-CONTROL-IMPLEMENTATION SCRIPT CODE.

!User-editable data

!STAGE 1

EnergyManagementSystem:ProgramCallingManager,

SetInputData,
BeginTimestepBeforePredictor,
SetInputData;

EnergyManagementSystem:Program,

SetInputData,
!Values as categorical data
!Comfort standard to be used: 0=CTE; 1 = EN16798-1; 2 = ASHRAE 55; 3 = JPN
set AdapStand = 3,
!Category or acceptability to be used: 1 = CAT I; 2 = CAT II; 3 = CAT III; 80 = 80% ACCEPT; 90 = 90% ACCEPT
set CAT = 3,
!Comfort mode to be used: 0 = Static; 1 = OUT-CTE; 2 = OUT-SENXXXX/SASHRAE55; 3 = OUT-AENXXXX/AASHRAE55
set ComfMod = 3,
!HVAC mode to be used: 0 = Fully Air-conditioned; 1 = Naturally ventilated; 2 = Mixed mode
set HVACmode = 2,
!Natural ventilation control to be used: 0 = Ventilates above comfort temperature; 1 = Ventilates above upper comfort limit
set VentCtrl = 0,
!Values as numerical data
!This values will be summed to the ventilation setpoint. Only used when VentCtrl = 0
set VSToffset = 0,
!AHST - MinOToffset will be the minimum outdoor temperature for ventilation, therefore it must be positive. Only used if HVACmode = 2.
set MinOToffset = 15,
!Maximum wind speed for ventilation.
set MaxWindSpeed = 6,
!This value will be summed to the cooling setpoint. It must be negative or 0.
set ACSTtol = -0.25,
!This value will be summed to the heating setpoint. It must be positive or 0.
set AHSTtol = 0.25;

!Not to be modified from here on.

!Declaration of global variables

EnergyManagementSystem:GlobalVariable,

!Adaptive Cooling Setpoint Temperature
ACST,
!Adaptive Heating Setpoint Temperature
AHST,
!Adaptive Cooling Setpoint Temperature without tolerance (i.e upper comfort limit)
ACSTnoTol,
!Adaptive Heating Setpoint Temperature without tolerance (i.e lower comfort limit)
AHSTnoTol,
!Comfort standard used
AdapStand,
!Adaptive Cooling Setpoint Temperature applicability upper limit
ACSTaul,
!Adaptive Cooling Setpoint Temperature applicability lower limit
ACSTall,
!Adaptive Heating Setpoint Temperature applicability upper limit
AHSTaul,
!Adaptive Heating Setpoint Temperature applicability lower limit
AHSTall,
!Category or acceptability level used
CAT,
!Offset of ACST from comfort temperature
ACSToffset,
!Offset of AHST from comfort temperature
AHSToffset,
!Comfort mode used
ComfMod,
!Comfort or neutral temperature
ComfTemp,
!ACST tolerance
ACSTtol,
!AHST tolerance
AHSTtol,

!Comfort hour
 ComfHours_Block1_Zone1,
 !Applicable but uncomfortable hour by heat
 DiscomfAppHotHours_Block1_Zone1,
 !Applicable but uncomfortable hour by cold
 DiscomfAppColdHours_Block1_Zone1,
 !Not applicable hour by heat
 DiscomfNonAppHotHours_Block1_Zone1,
 !Not applicable hour by cold
 DiscomfNonAppColdHours_Block1_Zone1,
 !Comfort hour (no applicability considered)
 ComfHoursNoApp_Block1_Zone1,
 !Ventilation hour
 VentHours_Block1_Zone1,
 !Ventilation setpoint temperature
 VST,
 !Offset for the VST
 VSToffset,
 !Maximum wind speed for ventilation
 MaxWindSpeed,
 !Natural ventilation control mode
 VentCtrl,
 !HVAC mode operation
 HVACmode,
 !Minimum outdoor temperature for natural ventilation
 MinOutTemp,
 !Offset for the MinOutTemp
 MinOToffset;

!Sensors
 !Running Mean Outdoor Temperature
 EnergyManagementSystem:Sensor,
 RMOT,
 People Block1:Zone1,
 Zone Thermal Comfort CEN 15251 Adaptive Model Running Average Outdoor Air Temperature;

!Prevailing Mean Outdoor Temperature
 EnergyManagementSystem:Sensor,
 PMOT,
 People Block1:Zone1,
 Zone Thermal Comfort ASHRAE 55 Adaptive Model Running Average Outdoor Air Temperature;

!Zone-related sensors: Block1:Zone1
 !Operative temperature
 EnergyManagementSystem:Sensor,
 Block1_Zone1_OpT,
 Block1:Zone1,
 Zone Operative Temperature;

!Cooling rate
 EnergyManagementSystem:Sensor,
 Block1_Zone1_CoolCoil,
 Block1:Zone1 VRF Indoor Unit DX Cooling Coil,
 Cooling Coil Total Cooling Rate;

!Heating rate
 EnergyManagementSystem:Sensor,
 Block1_Zone1_HeatCoil,
 Block1:Zone1 VRF Indoor Unit DX Heating Coil,
 Heating Coil Heating Rate;

!Wind speed
 EnergyManagementSystem:Sensor,
 Block1_Zone1_WindSpeed,
 Block1:Zone1,
 Zone Outdoor Air Wind Speed;

!Outdoor temperature
 EnergyManagementSystem:Sensor,
 Block1_Zone1_OutT,
 Block1:Zone1,

Zone Outdoor Air Drybulb Temperature;

!Windows-related sensors: Block1_Zone1_Wall_4_0_0_0_0_Win
!Operative temperature

EnergyManagementSystem:Sensor,
Block1_Zone1_Wall_4_0_0_0_0_Win_OpT,
Block1:Zone1,
Zone Operative Temperature;

!Cooling rate

EnergyManagementSystem:Sensor,
Block1_Zone1_Wall_4_0_0_0_0_Win_CoolCoil,
Block1:Zone1 VRF Indoor Unit DX Cooling Coil,
Cooling Coil Total Cooling Rate;

!Heating rate

EnergyManagementSystem:Sensor,
Block1_Zone1_Wall_4_0_0_0_0_Win_HeatCoil,
Block1:Zone1 VRF Indoor Unit DX Heating Coil,
Heating Coil Heating Rate;

!Wind speed

EnergyManagementSystem:Sensor,
Block1_Zone1_Wall_4_0_0_0_0_Win_WindSpeed,
Block1:Zone1,
Zone Outdoor Air Wind Speed;

!Outdoor temperature

EnergyManagementSystem:Sensor,
Block1_Zone1_Wall_4_0_0_0_0_Win_OutT,
Block1:Zone1,
Zone Outdoor Air Drybulb Temperature;

!Actuators

EnergyManagementSystem:Actuator,
!Actuator name
FORSCRIPT_ACST_Sch_Block1_Zone1,
!Schedule actuated: Cooling setpoint schedule
FORSCRIPT_ACST_Block1_Zone1,
Schedule:Compact,
Schedule Value;

EnergyManagementSystem:Actuator,
!Actuator name
FORSCRIPT_AHST_Sch_Block1_Zone1,
!Schedule actuated: Heating setpoint schedule
FORSCRIPT_AHST_Block1_Zone1,
Schedule:Compact,
Schedule Value;

EnergyManagementSystem:Actuator,
!Actuator name
Block1_Zone1_Wall_4_0_0_0_0_Win_VentOpenFact,
!Window actuated
Block1:Zone1_Wall_4_0_0_0_0_Win,
AirFlow Network Window/Door Opening,
!Field actuated of the window
Venting Opening Factor;

ISTAGE 2

EnergyManagementSystem:ProgramCallingManager,
SetComfTemp,
BeginTimestepBeforePredictor,
SetComfTemp;

EnergyManagementSystem:Program,
SetComfTemp,
if AdapStand == 1,
set ComfTemp = RMOT*0.33+18.8,

```

elseif AdapStand == 2,
    set ComfTemp = PMOT*0.31+17.8,
elseif AdapStand == 3,
    set ComfTemp = PMOT*0.48+14.4,
endif;

```

```

EnergyManagementSystem:ProgramCallingManager,
SetAppLimits,
BeginTimestepBeforePredictor,
SetAppLimits;

```

```

EnergyManagementSystem:Program,
SetAppLimits,
!Values to be updated in case of new standard revisions
If AdapStand == 1,
    set ACSTaul = 30,
    set ACSTall = 10,
    set AHSTaul = 30,
    set AHSTall = 10,
elseif AdapStand == 2,
    set ACSTaul = 33.5,
    set ACSTall = 10,
    set AHSTaul = 33.5,
    set AHSTall = 10,
elseif AdapStand == 3,
    set ACSTaul = 30,
    set ACSTall = 5,
    set AHSTaul = 30,
    set AHSTall = 5,
else,
    set ACSTaul = 50,
    set ACSTall = 50,
    set AHSTaul = 50,
    set AHSTall = 50,
endif;

```

```

EnergyManagementSystem:ProgramCallingManager,
ApplyCAT,
BeginTimestepBeforePredictor,
ApplyCAT;

```

```

EnergyManagementSystem:Program,
ApplyCAT,
if (AdapStand == 1 ),
    if (CAT == 1),
        set ACSToffset = 2,
        set AHSToffset = -3,
    elseif (CAT == 2),
        set ACSToffset = 3,
        set AHSToffset = -4,
    elseif (CAT == 3),
        set ACSToffset = 4,
        set AHSToffset = -5,
    endif,
elseif (AdapStand == 2 ) || (AdapStand == 3),
    if (CAT == 90),
        set ACSToffset = 2.5,
        set AHSToffset = -2.5,
    elseif (CAT == 80),
        set ACSToffset = 3.5,
        set AHSToffset = -3.5,
    endif,
endif;

```

```

EnergyManagementSystem:ProgramCallingManager,
SetAST,
BeginTimestepBeforePredictor,
SetAST;

```

```

EnergyManagementSystem:Program,
SetAST,

```

```

if (AdapStand == 0) && (CurrentTime < 7),
    set ACST = 27+ACSTtol,
    set AHST = 17+AHSTtol,
elseif (AdapStand == 0) && (CurrentTime < 15),
    set ACST = 25,
    set AHST = 20+AHSTtol,
elseif (AdapStand == 0) && (CurrentTime < 23),
    set ACST = 25+ACSTtol,
    set AHST = 20+AHSTtol,
elseif (AdapStand == 0) && (CurrentTime < 24),
    set ACST = 27+ACSTtol,
    set AHST = 17+AHSTtol,
endif,

if (AdapStand == 1) && (ComfMod == 0),
    if (DayOfYear >= 121) && (DayOfYear < 274),
        if (CAT==1),
            set ACST = 25.5+ACSTtol,
        elseif (CAT==2),
            set ACST = 26+ACSTtol,
        elseif (CAT==3),
            set ACST = 27+ACSTtol,
        endif,
    else,
        if (CAT==1),
            set ACST = 25+ACSTtol,
        elseif (CAT==2),
            set ACST = 25+ACSTtol,
        elseif (CAT==3),
            set ACST = 25+ACSTtol,
        endif,
    endif,
endif,

if (AdapStand == 1) && (ComfMod == 0),
    if (DayOfYear >= 121) && (DayOfYear < 274),
        if (CAT==1),
            set AHST = 23.5+AHSTtol,
        elseif (CAT==2),
            set AHST = 23+AHSTtol,
        elseif (CAT==3),
            set AHST = 22+AHSTtol,
        endif,
    else,
        if (CAT==1),
            set AHST = 21+AHSTtol,
        elseif (CAT==2),
            set AHST = 20+AHSTtol,
        elseif (CAT==3),
            set AHST = 18+AHSTtol,
        endif,
    endif,
endif,

if (AdapStand == 1) && (ComfMod == 1) && (RMOT >= ACSTall) && (RMOT <= ACSTaul),
    set ACST = RMOT*0.33+18.8+ACSToffset+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 1) && (CurrentTime < 7),
    set ACST = 27+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 1) && (CurrentTime < 15),
    set ACST = 50,
elseif (AdapStand == 1) && (ComfMod == 1) && (CurrentTime < 23),
    set ACST = 25+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 1) && (CurrentTime < 24),
    set ACST = 27+ACSTtol,
endif,

if (AdapStand == 1) && (ComfMod == 1) && (RMOT >= AHSTall) && (RMOT <= AHSTaul),
    set AHST = RMOT*0.33+18.8+AHSToffset+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 1) && (CurrentTime < 7),
    set AHST = 17+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 1) && (CurrentTime < 23),

```

```

        set AHST = 20+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 1) && (CurrentTime < 24),
        set AHST = 17+AHSTtol,
endif,

if (AdapStand == 1) && (ComfMod == 2) && (RMOT >= ACSTall) && (RMOT <= ACSTaul),
        set ACST = RMOT*0.33+18.8+ACSToffset+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT < ACSTall) && (CAT==1),
        set ACST = 25+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT > ACSTaul) && (CAT==1),
        set ACST = 25.5+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT < ACSTall) && (CAT==2),
        set ACST = 25+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT > ACSTaul) && (CAT==2),
        set ACST = 26+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT < ACSTall) && (CAT==3),
        set ACST = 25+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT > ACSTaul) && (CAT==3),
        set ACST = 27+ACSTtol,
endif,

if (AdapStand == 1) && (ComfMod == 2) && (RMOT >= AHSTall) && (RMOT <= AHSTaul),
        set AHST = RMOT*0.33+18.8+AHSToffset+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT < AHSTall) && (CAT==1),
        set AHST = 21+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT > AHSTaul) && (CAT==1),
        set AHST = 23.5+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT < AHSTall) && (CAT==2),
        set AHST = 20+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT > AHSTaul) && (CAT==2),
        set AHST = 23+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT < AHSTall) && (CAT==3),
        set AHST = 18+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 2) && (RMOT > AHSTaul) && (CAT==3),
        set AHST = 22+AHSTtol,
endif,

if (AdapStand == 1) && (ComfMod == 3) && (RMOT >= ACSTall) && (RMOT <= ACSTaul),
        set ACST = RMOT*0.33+18.8+ACSToffset+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 3) && (RMOT < ACSTall),
        set ACST = ACSTall*0.33+18.8+ACSToffset+ACSTtol,
elseif (AdapStand == 1) && (ComfMod == 3) && (RMOT > ACSTaul),
        set ACST = ACSTaul*0.33+18.8+ACSToffset+ACSTtol,
endif,

if (AdapStand == 1) && (ComfMod == 3) && (RMOT >= AHSTall) && (RMOT <= AHSTaul),
        set AHST = RMOT*0.33+18.8+AHSToffset+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 3) && (RMOT < AHSTall),
        set AHST = AHSTall*0.33+18.8+AHSToffset+AHSTtol,
elseif (AdapStand == 1) && (ComfMod == 3) && (RMOT > AHSTaul),
        set AHST = AHSTaul*0.33+18.8+AHSToffset+AHSTtol,
endif,

if (AdapStand == 2) && (ComfMod == 0),
        if (DayOfYear >= 121) && (DayOfYear < 274),
                if (CAT==80),
                        set ACST = 28.35+ACSTtol,
                elseif (CAT==90),
                        set ACST = 27.42+ACSTtol,
                endif,
        else,
                if (CAT==80),
                        set ACST = 26.35+ACSTtol,
                elseif (CAT==90),
                        set ACST = 25.09+ACSTtol,
                endif,
        endif,
endif,

if (AdapStand == 2) && (ComfMod == 0),
        if (DayOfYear >= 121) && (DayOfYear < 274),

```

```

        if (CAT==80),
            set AHST = 23.78+AHSTtol,
        elseif (CAT==90),
            set AHST = 24.74+AHSTtol,
        endif,
    else,
        if (CAT==80),
            set AHST = 20.1+AHSTtol,
        elseif (CAT==90),
            set AHST = 21.44+AHSTtol,
        endif,
    endif,
endif,

if (AdapStand == 2) && (ComfMod == 1) && (PMOT >= ACSTall) && (PMOT <= ACSTaul),
    set ACST = PMOT*0.31+17.8+ACSToffset+ACSTtol,
elseif (AdapStand == 2) && (ComfMod == 1) && (CurrentTime < 7),
    set ACST = 27+ACSTtol,
elseif (AdapStand == 2) && (ComfMod == 1) && (CurrentTime < 15),
    set ACST = 50,
elseif (AdapStand == 2) && (ComfMod == 1) && (CurrentTime < 23),
    set ACST = 25+ACSTtol,
elseif (AdapStand == 2) && (ComfMod == 1) && (CurrentTime < 24),
    set ACST = 27+ACSTtol,
endif,

if (AdapStand == 2) && (ComfMod == 1) && (PMOT >= AHSTall) && (PMOT <= AHSTaul),
    set AHST = PMOT*0.31+17.8+AHSToffset+AHSTtol,
elseif (AdapStand == 2) && (ComfMod == 1) && (CurrentTime < 7),
    set AHST = 17+AHSTtol,
elseif (AdapStand == 2) && (ComfMod == 1) && (CurrentTime < 23),
    set AHST = 20+AHSTtol,
elseif (AdapStand == 2) && (ComfMod == 1) && (CurrentTime < 24),
    set AHST = 17+AHSTtol,
endif,

if (AdapStand == 2) && (ComfMod == 2),
    if (PMOT >= ACSTall) && (PMOT <= ACSTaul),
        set ACST = PMOT*0.31+17.8+ACSToffset+ACSTtol,
    elseif CAT==80,
        if PMOT < ACSTall,
            set ACST = 26.35+ACSTtol,
        elseif PMOT > ACSTaul,
            set ACST = 28.35+ACSTtol,
        endif,
    elseif CAT==90,
        if PMOT < ACSTall,
            set ACST = 25.09+ACSTtol,
        elseif PMOT > ACSTaul,
            set ACST = 27.42+ACSTtol,
        endif,
    endif,
endif,

if (AdapStand == 2) && (ComfMod == 2),
    if (PMOT >= AHSTall) && (PMOT <= AHSTaul),
        set AHST = PMOT*0.31+17.8+AHSToffset+AHSTtol,
    elseif CAT==80,
        if PMOT < AHSTall,
            set AHST = 20.1+AHSTtol,
        elseif PMOT > AHSTaul,
            set AHST = 23.78+AHSTtol,
        endif,
    elseif CAT==90,
        if PMOT < AHSTall,
            set AHST = 21.44+AHSTtol,
        elseif PMOT > AHSTaul,
            set AHST = 24.74+AHSTtol,
        endif,
    endif,
endif,
endif,

```

```

if (AdapStand == 2) && (ComfMod == 3) && (PMOT >= ACSTall) && (PMOT <= ACSTaul),
    set ACST = PMOT*0.31+17.8+ACSToffset+ACSTtol,
elseif (AdapStand == 2) && (ComfMod == 3) && (PMOT < ACSTall),
    set ACST = ACSTall*0.31+17.8+ACSToffset+ACSTtol,
elseif (AdapStand == 2) && (ComfMod == 3) && (PMOT > ACSTaul),
    set ACST = ACSTaul*0.31+17.8+ACSToffset+ACSTtol,
endif,

if (AdapStand == 2) && (ComfMod == 3) && (PMOT >= AHSTall) && (PMOT <= AHSTaul),
    set AHST = PMOT*0.31+17.8+AHSToffset+AHSTtol,
elseif (AdapStand == 2) && (ComfMod == 3) && (PMOT < AHSTall),
    set AHST = AHSTall*0.31+17.8+AHSToffset+AHSTtol,
elseif (AdapStand == 2) && (ComfMod == 3) && (PMOT > AHSTaul),
    set AHST = AHSTaul*0.31+17.8+AHSToffset+AHSTtol,
endif,

if (AdapStand == 3) && (ComfMod == 0),
    if (CAT==80),
        set ACST = 28+ACSTtol,
    elseif (CAT==90),
        set ACST = 27+ACSTtol,
    endif,
endif,

if (AdapStand == 3) && (ComfMod == 0),
    if (CAT==80),
        set AHST = 18+AHSTtol,
    elseif (CAT==90),
        set AHST = 19+AHSTtol,
    endif,
endif,

if (AdapStand == 3) && (ComfMod == 1),
    if (PMOT >= ACSTall) && (PMOT <= ACSTaul),
        set ACST = PMOT*0.48+14.4+ACSToffset+ACSTtol,
    elseif CAT==80,
        if PMOT < ACSTall,
            set ACST = 28+ACSTtol,
        elseif PMOT > ACSTaul,
            set ACST = 28+ACSTtol,
        endif,
    elseif CAT==90,
        if PMOT < ACSTall,
            set ACST = 27+ACSTtol,
        elseif PMOT > ACSTaul,
            set ACST = 27+ACSTtol,
        endif,
    endif,
endif,

if (AdapStand == 3) && (ComfMod == 1),
    if (PMOT >= AHSTall) && (PMOT <= AHSTaul),
        set AHST = PMOT*0.48+14.4+AHSToffset+AHSTtol,
    elseif CAT==80,
        if PMOT < AHSTall,
            set AHST = 18+AHSTtol,
        elseif PMOT > AHSTaul,
            set AHST = 18+AHSTtol,
        endif,
    elseif CAT==90,
        if PMOT < AHSTall,
            set AHST = 19+AHSTtol,
        elseif PMOT > AHSTaul,
            set AHST = 19+AHSTtol,
        endif,
    endif,
endif,

if (AdapStand == 3) && (ComfMod == 2),
    if (PMOT >= ACSTall) && (PMOT <= ACSTaul),

```



```

        set ACST = PMOT*0.48+14.4+ACSTOffset+ACSTtol,
    elseif CAT==80,
        if PMOT < ACSTall,
            set ACST = 26.35+ACSTtol,
        elseif PMOT > ACSTaul,
            set ACST = 28.35+ACSTtol,
        endif,
    elseif CAT==90,
        if PMOT < ACSTall,
            set ACST = 25.09+ACSTtol,
        elseif PMOT > ACSTaul,
            set ACST = 27.42+ACSTtol,
        endif,
    endif,
endif,

if (AdapStand == 3) && (ComfMod == 2),
    if (PMOT >= AHSTall) && (PMOT <= AHSTaul),
        set AHST = PMOT*0.48+14.4+AHSTOffset+AHSTtol,
    elseif CAT==80,
        if PMOT < AHSTall,
            set AHST = 20.1+AHSTtol,
        elseif PMOT > AHSTaul,
            set AHST = 23.78+AHSTtol,
        endif,
    elseif CAT==90,
        if PMOT < AHSTall,
            set AHST = 21.44+AHSTtol,
        elseif PMOT > AHSTaul,
            set AHST = 24.74+AHSTtol,
        endif,
    endif,
endif,

if (AdapStand == 3) && (ComfMod == 3) && (PMOT >= ACSTall) && (PMOT <= ACSTaul),
    set ACST = PMOT*0.48+14.4+ACSTOffset+ACSTtol,
elseif (AdapStand == 3) && (ComfMod == 3) && (PMOT < ACSTall),
    set ACST = ACSTall*0.48+14.4+ACSTOffset+ACSTtol,
elseif (AdapStand == 3) && (ComfMod == 3) && (PMOT > ACSTaul),
    set ACST = ACSTaul*0.48+14.4+ACSTOffset+ACSTtol,
endif,

if (AdapStand == 3) && (ComfMod == 3) && (PMOT >= AHSTall) && (PMOT <= AHSTaul),
    set AHST = PMOT*0.48+14.4+AHSTOffset+AHSTtol,
elseif (AdapStand == 3) && (ComfMod == 3) && (PMOT < AHSTall),
    set AHST = AHSTall*0.48+14.4+AHSTOffset+AHSTtol,
elseif (AdapStand == 3) && (ComfMod == 3) && (PMOT > AHSTaul),
    set AHST = AHSTaul*0.48+14.4+AHSTOffset+AHSTtol,
endif;

```

```

EnergyManagementSystem:ProgramCallingManager,
SetASTnoTol,
BeginTimestepBeforePredictor,
SetASTnoTol;

```

```

EnergyManagementSystem:Program,
SetASTnoTol,
set ACSTnoTol = ACST-ACSTtol,
set AHSTnoTol = AHST-AHSTtol;

```

```

EnergyManagementSystem:ProgramCallingManager,
SetVST,
BeginTimestepBeforePredictor,
SetVST;

```

```

EnergyManagementSystem:Program,
SetVST,
set MinOutTemp = AHST - MinOToffset,
if AdapStand == 0,
    if (CurrentTime < 7),
        set VST = (ACST+AHST)/2+VSToffset,
    endif,
endif,

```

```

elseif (CurrentTime < 15),
    set VST = 22.5+VSTOffset,
elseif (CurrentTime < 23),
    set VST = (ACST+AHST)/2+VSTOffset,
elseif (CurrentTime < 24),
    set VST = (ACST+AHST)/2+VSTOffset,
endif,
elseif AdapStand == 1,
    if (RMOT >= AHSTall) && (RMOT <= ACSTaul),
        set VST = ComfTemp+VSTOffset,
    else,
        set VST = (ACST+AHST)/2+VSTOffset,
    endif,
elseif AdapStand == 2 || AdapStand == 3,
    if (PMOT >= AHSTall) && (PMOT <= ACSTaul),
        set VST = ComfTemp+VSTOffset,
    else,
        set VST = (ACST+AHST)/2+VSTOffset,
    endif,
endif;

```

ISTAGE 3

EnergyManagementSystem:ProgramCallingManager,

```

CountHours_Block1_Zone1,
BeginTimestepBeforePredictor,
CountHours_Block1_Zone1;

```

EnergyManagementSystem:Program,

```

CountHours_Block1_Zone1,
if (AdapStand == 1),
    if (RMOT >= AHSTall) && (RMOT <= ACSTaul),
        if (Block1_Zone1_OpT <= ACSTnoTol),
            if (Block1_Zone1_OpT >= AHSTnoTol),
                set ComfHours_Block1_Zone1 = 1*ZoneTimeStep,
                set DiscomfAppHotHours_Block1_Zone1 = 0,
                set DiscomfAppColdHours_Block1_Zone1 = 0,
                set DiscomfNonAppHotHours_Block1_Zone1 = 0,
                set DiscomfNonAppColdHours_Block1_Zone1 = 0,
            endif,
        elseif (Block1_Zone1_OpT > ACSTnoTol),
            set ComfHours_Block1_Zone1 = 0,
            set DiscomfAppHotHours_Block1_Zone1 = 1*ZoneTimeStep,
            set DiscomfAppColdHours_Block1_Zone1 = 0,
            set DiscomfNonAppHotHours_Block1_Zone1 = 0,
            set DiscomfNonAppColdHours_Block1_Zone1 = 0,
        elseif (Block1_Zone1_OpT < AHSTnoTol),
            set ComfHours_Block1_Zone1 = 0,
            set DiscomfAppHotHours_Block1_Zone1 = 0,
            set DiscomfAppColdHours_Block1_Zone1 = 1*ZoneTimeStep,
            set DiscomfNonAppHotHours_Block1_Zone1 = 0,
            set DiscomfNonAppColdHours_Block1_Zone1 = 0,
        endif,
    elseif (RMOT > ACSTaul),
        set ComfHours_Block1_Zone1 = 0,
        set DiscomfAppHotHours_Block1_Zone1 = 0,
        set DiscomfAppColdHours_Block1_Zone1 = 0,
        set DiscomfNonAppHotHours_Block1_Zone1 = 1*ZoneTimeStep,
        set DiscomfNonAppColdHours_Block1_Zone1 = 0,
    elseif (RMOT < AHSTall),
        set ComfHours_Block1_Zone1 = 0,
        set DiscomfAppHotHours_Block1_Zone1 = 0,
        set DiscomfAppColdHours_Block1_Zone1 = 0,
        set DiscomfNonAppHotHours_Block1_Zone1 = 0,
        set DiscomfNonAppColdHours_Block1_Zone1 = 1*ZoneTimeStep,
    endif,
elseif (AdapStand == 2) || (AdapStand == 3),
    if (PMOT >= AHSTall) && (PMOT <= ACSTaul),
        if (Block1_Zone1_OpT <= ACSTnoTol),
            if (Block1_Zone1_OpT >= AHSTnoTol),
                set ComfHours_Block1_Zone1 = 1*ZoneTimeStep,

```

```

        set DiscomfAppHotHours_Block1_Zone1 = 0,
        set DiscomfAppColdHours_Block1_Zone1 = 0,
        set DiscomfNonAppHotHours_Block1_Zone1 = 0,
        set DiscomfNonAppColdHours_Block1_Zone1 = 0,
    endif,
elseif (Block1_Zone1_OpT > ACSTnoTol),
    set ComfHours_Block1_Zone1 = 0,
    set DiscomfAppHotHours_Block1_Zone1 = 1*ZoneTimeStep,
    set DiscomfAppColdHours_Block1_Zone1 = 0,
    set DiscomfNonAppHotHours_Block1_Zone1 = 0,
    set DiscomfNonAppColdHours_Block1_Zone1 = 0,
elseif (Block1_Zone1_OpT < AHSTnoTol),
    set ComfHours_Block1_Zone1 = 0,
    set DiscomfAppHotHours_Block1_Zone1 = 0,
    set DiscomfAppColdHours_Block1_Zone1 = 1*ZoneTimeStep,
    set DiscomfNonAppHotHours_Block1_Zone1 = 0,
    set DiscomfNonAppColdHours_Block1_Zone1 = 0,
endif,
elseif (PMOT > ACSTaul),
    set ComfHours_Block1_Zone1 = 0,
    set DiscomfAppHotHours_Block1_Zone1 = 0,
    set DiscomfAppColdHours_Block1_Zone1 = 0,
    set DiscomfNonAppHotHours_Block1_Zone1 = 1*ZoneTimeStep,
    set DiscomfNonAppColdHours_Block1_Zone1 = 0,
elseif (PMOT < AHSTall),
    set ComfHours_Block1_Zone1 = 0,
    set DiscomfAppHotHours_Block1_Zone1 = 0,
    set DiscomfAppColdHours_Block1_Zone1 = 0,
    set DiscomfNonAppHotHours_Block1_Zone1 = 0,
    set DiscomfNonAppColdHours_Block1_Zone1 = 1*ZoneTimeStep,
endif,
endif;

```

EnergyManagementSystem:ProgramCallingManager,

```

CountHoursNoApp_Block1_Zone1,
BeginTimestepBeforePredictor,
CountHoursNoApp_Block1_Zone1;

```

EnergyManagementSystem:Program,

```

CountHoursNoApp_Block1_Zone1,
if (Block1_Zone1_OpT <= ACSTnoTol),
    if (Block1_Zone1_OpT >= AHSTnoTol),
        set ComfHoursNoApp_Block1_Zone1 = 1*ZoneTimeStep,
    else,
        set ComfHoursNoApp_Block1_Zone1 = 0,
    endif,
else,
    set ComfHoursNoApp_Block1_Zone1 = 0,
endif;

```

EnergyManagementSystem:ProgramCallingManager,

```

ApplyAST_Block1_Zone1,
BeginTimestepBeforePredictor,
ApplyAST_Block1_Zone1;

```

EnergyManagementSystem:Program,

```

ApplyAST_Block1_Zone1,

```

!First if-block

```

if (Block1_Zone1_OpT > VST) && (Block1_Zone1_OutT < VST),
    if Block1_Zone1_CoolCoil == 0,
        if Block1_Zone1_HeatCoil == 0,
            if (Block1_Zone1_OpT < ACST) && (Block1_Zone1_OutT > MinOutTemp),
                if Block1_Zone1_WindSpeed <= MaxWindSpeed,
                    set Ventilates_HVACmode2_Block1_Zone1 = 1,
                else,
                    set Ventilates_HVACmode2_Block1_Zone1 = 0,
            endif,
        endif,
    endif,
endif;

```

```

else,
    set Ventilates_HVACmode2_Block1_Zone1 = 0,
endif,
else,
    set Ventilates_HVACmode2_Block1_Zone1 = 0,
endif,
else,
    set Ventilates_HVACmode2_Block1_Zone1 = 0,
endif,
else,
    set Ventilates_HVACmode2_Block1_Zone1 = 0,
endif,
endif,
!Second if-block
if Block1_Zone1_OutT < Block1_Zone1_OpT,
    if Block1_Zone1_OutT > MinOutTemp,
        if Block1_Zone1_OpT > VST,
            if Block1_Zone1_WindSpeed <= MaxWindSpeed,
                set Ventilates_HVACmode1_Block1_Zone1 = 1,
            else,
                set Ventilates_HVACmode1_Block1_Zone1 = 0,
            endif,
        else,
            set Ventilates_HVACmode1_Block1_Zone1 = 0,
        endif,
    else,
        set Ventilates_HVACmode1_Block1_Zone1 = 0,
    endif,
else,
    set Ventilates_HVACmode1_Block1_Zone1 = 0,
endif,
endif,
elseif VentCtrl == 1,
    if Block1_Zone1_OutT < Block1_Zone1_OpT,
        if Block1_Zone1_OutT > MinOutTemp,
            if Block1_Zone1_OpT > ACSTnoTol,
                if Block1_Zone1_WindSpeed <= MaxWindSpeed,
                    set Ventilates_HVACmode1_Block1_Zone1 = 1,
                else,
                    set Ventilates_HVACmode1_Block1_Zone1 = 0,
                endif,
            else,
                set Ventilates_HVACmode1_Block1_Zone1 = 0,
            endif,
        else,
            set Ventilates_HVACmode1_Block1_Zone1 = 0,
        endif,
    else,
        set Ventilates_HVACmode1_Block1_Zone1 = 0,
    endif,
endif,
!Third if-block
if HVACmode == 0,
    set FORSCRIPT_ACST_Sch_Block1_Zone1 = ACST,
    set FORSCRIPT_AHST_Sch_Block1_Zone1 = AHST,
elseif HVACmode == 1,
    Set FORSCRIPT_ACST_Sch_Block1_Zone1 = 100,
    Set FORSCRIPT_AHST_Sch_Block1_Zone1 = -100,
    if Ventilates_HVACmode1_Block1_Zone1 == 1,
        set VentHours_Block1_Zone1 = 1,
    else,
        set VentHours_Block1_Zone1 = 0,
    endif,
elseif HVACmode == 2,
    if Ventilates_HVACmode2_Block1_Zone1 == 1,
        set VentHours_Block1_Zone1 = 1,
    elseif Ventilates_HVACmode2_Block1_Zone1 == 0,
        set VentHours_Block1_Zone1 = 0,
        set FORSCRIPT_ACST_Sch_Block1_Zone1 = ACST,
        set FORSCRIPT_AHST_Sch_Block1_Zone1 = AHST,
    endif,
endif;

```

EnergyManagementSystem:ProgramCallingManager,

SetWindowOperation_Block1_Zone1_Wall_4_0_0_0_0_Win,
BeginTimestepBeforePredictor,
SetWindowOperation_Block1_Zone1_Wall_4_0_0_0_0_Win;

EnergyManagementSystem:Program,

SetWindowOperation_Block1_Zone1_Wall_4_0_0_0_0_Win,
!First if-block

```
if (Block1_Zone1_Wall_4_0_0_0_0_Win_OpT>VST)&&(Block1_Zone1_Wall_4_0_0_0_0_Win_OutT < VST),
    if Block1_Zone1_Wall_4_0_0_0_0_Win_CoolCoil==0,
        if Block1_Zone1_Wall_4_0_0_0_0_Win_HeatCoil==0,
            if
                (Block1_Zone1_Wall_4_0_0_0_0_Win_OpT<ACST)&&(Block1_Zone1_Wall_4_0_0_0_0_Win_OutT>MinOutTemp),
                    if Block1_Zone1_Wall_4_0_0_0_0_Win_WindSpeed <= MaxWindSpeed,
                        set Ventilates_HVACmode2_Block1_Zone1_Wall_4_0_0_0_0_Win = 1,
                    else,
                        set Ventilates_HVACmode2_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
                    endif,
                else,
                    set Ventilates_HVACmode2_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
                endif,
            else,
                set Ventilates_HVACmode2_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
            endif,
        else,
            set Ventilates_HVACmode2_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
        endif,
    else,
        set Ventilates_HVACmode2_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
    endif,
endif,
```

!Second if-block

```
if VentCtrl == 0,
    if Block1_Zone1_Wall_4_0_0_0_0_Win_OutT < Block1_Zone1_Wall_4_0_0_0_0_Win_OpT,
        if Block1_Zone1_Wall_4_0_0_0_0_Win_OutT > MinOutTemp,
            if Block1_Zone1_Wall_4_0_0_0_0_Win_OpT > VST,
                if Block1_Zone1_Wall_4_0_0_0_0_Win_WindSpeed <= MaxWindSpeed,
                    set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 1,
                else,
                    set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
                endif,
            else,
                set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
            endif,
        else,
            set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
        endif,
    else,
        set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
    endif,
elseif VentCtrl == 1,
    if Block1_Zone1_Wall_4_0_0_0_0_Win_OutT < Block1_Zone1_Wall_4_0_0_0_0_Win_OpT,
        if Block1_Zone1_Wall_4_0_0_0_0_Win_OutT > MinOutTemp,
            if Block1_Zone1_Wall_4_0_0_0_0_Win_OpT > ACSTnoTol,
                if Block1_Zone1_Wall_4_0_0_0_0_Win_WindSpeed <= MaxWindSpeed,
                    set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 1,
                else,
                    set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
                endif,
            else,
                set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
            endif,
        else,
            set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
        endif,
    else,
        set Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_Win = 0,
    endif,
endif,
```

!Third if-block

```

if HVACmode == 0,
    set Block1_Zone1_Wall_4_0_0_0_0_0_Win_VentOpenFact = 0,
elseif HVACmode == 1,
    if Ventilates_HVACmode1_Block1_Zone1_Wall_4_0_0_0_0_0_Win == 1,
        set Block1_Zone1_Wall_4_0_0_0_0_0_Win_VentOpenFact = 1,
    else,
        set Block1_Zone1_Wall_4_0_0_0_0_0_Win_VentOpenFact = 0,
    endif,
elseif HVACmode == 2,
    if Ventilates_HVACmode2_Block1_Zone1_Wall_4_0_0_0_0_0_Win == 1,
        set Block1_Zone1_Wall_4_0_0_0_0_0_Win_VentOpenFact = 1,
    else,
        set Block1_Zone1_Wall_4_0_0_0_0_0_Win_VentOpenFact = 0,
    endif,
endif;

```

ISTAGE 4

EnergyManagementSystem:OutputVariable,

Comfort Temperature,
 ComfTemp,
 Averaged,
 ZoneTimestep,
 ,
 C;

EnergyManagementSystem:OutputVariable,

Adaptive Cooling Setpoint Temperature,
 ACST,
 Averaged,
 ZoneTimestep,
 ,
 C;

EnergyManagementSystem:OutputVariable,

Adaptive Heating Setpoint Temperature,
 AHST,
 Averaged,
 ZoneTimestep,
 ,
 C;

EnergyManagementSystem:OutputVariable,

Adaptive Cooling Setpoint Temperature_No Tolerance,
 ACSTnoTol,
 Averaged,
 ZoneTimestep,
 ,
 C;

EnergyManagementSystem:OutputVariable,

Adaptive Heating Setpoint Temperature_No Tolerance,
 AHSTnoTol,
 Averaged,
 ZoneTimestep,
 ,
 C;

EnergyManagementSystem:OutputVariable,

Comfortable Hours_No Applicability_Block1_Zone1 (summed),
 ComfHoursNoApp_Block1_Zone1,
 Summed,
 ZoneTimestep,
 ,
 h;

EnergyManagementSystem:OutputVariable,

Comfortable Hours_Block1_Zone1 (summed),
 ComfHours_Block1_Zone1,
 Summed,
 ZoneTimestep,

,
h;

EnergyManagementSystem:OutputVariable,
Discomfortable Applicable Hot Hours_Block1_Zone1 (summed),
DiscomfAppHotHours_Block1_Zone1,
Summed,
ZoneTimestep,
,
h;

EnergyManagementSystem:OutputVariable,
Discomfortable Applicable Cold Hours_Block1_Zone1 (summed),
DiscomfAppColdHours_Block1_Zone1,
Summed,
ZoneTimestep,
,
h;

EnergyManagementSystem:OutputVariable,
Discomfortable Non Applicable Hot Hours_Block1_Zone1 (summed),
DiscomfNonAppHotHours_Block1_Zone1,
Summed,
ZoneTimestep,
,
h;

EnergyManagementSystem:OutputVariable,
Discomfortable Non Applicable Cold Hours_Block1_Zone1 (summed),
DiscomfNonAppColdHours_Block1_Zone1,
Summed,
ZoneTimestep,
,
h;

EnergyManagementSystem:OutputVariable,
Ventilation Hours_Block1_Zone1 (summed),
VentHours_Block1_Zone1,
Summed,
ZoneTimestep,
,
h;

EnergyManagementSystem:OutputVariable,
Ventilation Setpoint Temperature,
VST,
Averaged,
ZoneTimestep,
,
C;

EnergyManagementSystem:OutputVariable,
Minimum Outdoor Temperature for mixed mode ventilation,
MinOutTemp,
Averaged,
ZoneTimestep,
,
C;

Output:Variable,
*, Zone Thermostat Operative Temperature, Hourly;
Output:Variable,
*, Adaptive Cooling Setpoint Temperature, Hourly;
Output:Variable,
*, Adaptive Heating Setpoint Temperature, Hourly;
Output:Variable,
*, Adaptive Cooling Setpoint Temperature_No Tolerance, Hourly;
Output:Variable,
*, Adaptive Heating Setpoint Temperature_No Tolerance, Hourly;
Output:Variable,

*, Comfort Temperature, Hourly;
 Output:Variable,
 *, Ventilation Setpoint Temperature, Hourly;
 Output:Variable,
 *, Minimum Outdoor Temperature for mixed mode ventilation, Hourly;
 Output:Variable,
 Environment, Site Outdoor Air Drybulb Temperature, Hourly;
 Output:Variable,
 *, Zone Thermal Comfort CEN 15251 Adaptive Model Running Average Outdoor Air Temperature, Hourly;
 Output:Variable,
 *, Zone Operative Temperature, Hourly;
 Output:Variable,
 *, Comfortable Hours_No Applicability_Block1_Zone1 (summed), Hourly;
 Output:Variable,
 *, Comfortable Hours_Block1_Zone1 (summed), Hourly;
 Output:Variable,
 *, Discomfortable Applicable Hot Hours_Block1_Zone1 (summed), Hourly;
 Output:Variable,
 *, Discomfortable Applicable Cold Hours_Block1_Zone1 (summed), Hourly;
 Output:Variable,
 *, Discomfortable Non Applicable Hot Hours_Block1_Zone1 (summed), Hourly;
 Output:Variable,
 *, Discomfortable Non Applicable Cold Hours_Block1_Zone1 (summed), Hourly;
 Output:Variable,
 *, Ventilation Hours_Block1_Zone1 (summed), Hourly;
 Output:Variable,
 Block1:Zone1, Zone Operative Temperature, Hourly;
 Output:Variable,
 FORSCRIPT_AHST_Block1_Zone1, Schedule Value, Hourly;
 Output:Variable,
 FORSCRIPT_ACST_Block1_Zone1, Schedule Value, Hourly;
 Output:Variable,
 Ventilation Hours_Block1_Zone1 (summed), Schedule Value, Hourly;
 Output:Variable,
 *, Zone Floor Area_Block1_Zone1, Hourly;
 Output:Variable,
 *, Zone Air Volume_Block1_Zone1, Hourly;
 Output:Variable,
 *, AFN Surface Venting Window or Door Opening Factor, Hourly;
 Output:Variable,
 *, AFN Zone Infiltration Air Change Rate, Hourly;
 Output:Variable,
 *, AFN Zone Infiltration Volume, Hourly;
 Output:Variable,
 *, VRF Heat Pump Cooling Electric Energy, Hourly;
 Output:Variable,
 *, VRF Heat Pump Heating Electric Energy, Hourly;
 Output:Variable,
 *, Cooling Coil Total Cooling Rate, Hourly;
 Output:Variable,
 *, Heating Coil Heating Rate, Hourly;
 Output:Variable,
 Block1:Zone1 VRF Indoor Unit DX Cooling Coil,Cooling Coil Total Cooling Rate, Hourly;
 Output:Variable,
 Block1:Zone1 VRF Indoor Unit DX Heating Coil,Heating Coil Heating Rate, Hourly;
 Output:Variable,
 Block1:Zone1, Zone Outdoor Air Wind Speed, Hourly;
 Output:Variable,
 Block1:Zone1, Zone Outdoor Air Drybulb Temperature, Hourly;