Users Manual - OpenStudio Example EMS Measures

Presented by Performance Systems Development to:

National Renewable Energy Lab

Attn: Dr. Brian Ball

July 19, 2018

Performance Systems Development of New York, LLC.
124 Brindley Street, Ithaca, NY 14850
www.psdconsulting.com

Contents

Background

PSD has developed, tested and documented a set of OpenStudio measures which demonstrate the use of EnergyPlus Energy Management System (EMS) objects via OpenStudio API calls. The measures were developed and tested using OpenStudio v2.6.0.

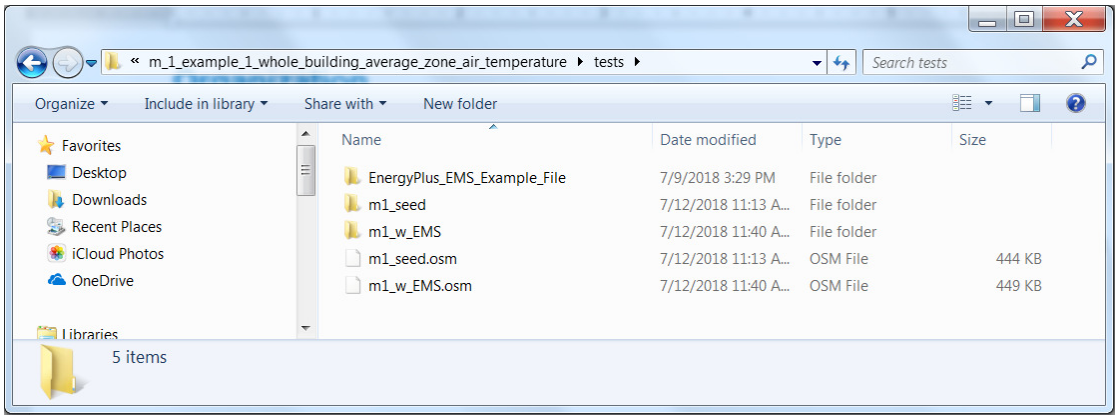
The work product includes 11 separate and complete OpenStudio measures, defined on pages 63 – 113 of EnergyPlus v8.9.0 Documentation: "Application Guide for EMS".

1) m_1_example_1_whole_building_average_zone_air_temperature
2) m_2_example_2_traditional_setpoint_and_availability_managers
3) m_4_example_4_halt_program_based_on_constraint
4) m_5_example_5_computed_schedule
5) m_7_example_7_constant_volume_purchased_air_system
6) m_8_example_8_system_sizing_with_discrete_package_sizes
7) m_9_example_9_demand_management
8) m_10_example_10_plant_loop_override_control
9) m_11_example_11_performance_curve_result_override
10) m_12_example_12_variable_refrigerant_flow_system_override
11) m_13_example_13_surface_construction_actuator_for_thermochromatic_window

Two of the thirteen EnergyPlus example EMS measures (Example 3 and Example 6) described in the EMS Applications Guide were unable to be translated into the OpenStudio environment due to limitations in the OS v2.6.0 API.

Organization

Each of the 11 OpenStudio EMS measures includes a subfolder named ""**tests**"", which includes a subfolder containing an example EnergyPlus EMS .idf file from EnergyPlus v8.9.0. Each **tests** folder also includes an OpenStudio file titled ""**m[x]_seed.osm**"". The OpenStudio EMS measures have been developed and tested to run against this file. The OpenStudio file named ""**m[x]_w_EMS.osm**"" is the result of applying the measures to the seed model, using the OpenStudio Application "Apply Measure Now" feature.



The OpenStudio EMS Examples section of this document describes in detail the “Measure Specification” document used to develop each measure. Each specification includes descriptions of user interactions (measure arguments), descriptions of error and warning massages, and pseudo-code used to develop and test the measure.

The measure.rb file associated with each measure includes copious supplemental documentation, such as code comments, remarks and inferences to external datasets.

Openstudio EMS Examples

M1 Example 1. Whole-Building Average Zone Air Temperature

Equivalent EnergyPlus v8.9.0 .idf file

None

DOE Prototype Buildings

This measure can be run against all models generated from the “Create DOE Prototype Building” measure available from the BCL.

Description

This EMS measure does not control anything – but rather uses EMS sensors and EMS internal variables to generate a single value representing whole-building average temperature, weighted by zonal volume. Only conditioned zones are included in determining the average temperature value. The calculation will be evaluated at each zone timestep. The measure demonstrates the creation and use of the following OpenStudio EMS objects.

1) EnergyManagementSystem:ProgramCallingManager
2) EnergyManagementSystem:Sensor
3) EnergyManagementSystem:InternalVariable
4) EnergyManagementSystem:OutputVariable
5) EnergyManagementSystem:Program
6) Output:EnergyManagementSystem

Modeler Description

This EMS measure will uses EnergyManagementSystem:Sensor objects (Zone Temperatures) and EnergyManagementSystem:InternalVariable objects (Zone Volume) to calculate a whole building volume-weighted average temperature at each zone timestep. The result is stored in an EnergyManagementSystem:OutputVariable object.

Measure Type

This measure will be created as an OpenStudio Measure tagged as “HVAC : HVAC Controls”.

Arguments

1) EMS Reporting:
    a. InternalVariableAvailabilityDictionaryReporting
        i. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
    b. EMSRuntimeLanguageDebugOutputLevel
        i. Choice List for level of EMS Reporting: (““None”“, ““ErrorsOnly”“, ““Verbose”“)
    c. ActuatorAvailabilityDictionaryReporting value
        i. Choice List for level of EMS Reporting: (““None”“, ““NotByUniqueNames”“, ““Verbose”“)

Error / Warning / NA Messages

1) An N/A message will be generated if the model has zero conditioned zones.

Measure Pseudocode

1) Get user arguments
2) Determine the number of and name of zones to include in the calculation
3) Loop through each zone
    a. If conditioned,
        i. Create new EMS Sensor object
        ii. Create EMS Internal Variable Object
4) Create new EMS Program Calling Manager Object
5) Create new Output Variable Object
6) Create new EMS Program Object
7) Create new EMS Output:Variable object
8) Configure OutputEnergyManagementSystem object with user arguments

Measure Testing

A PAT project was used to test the measure as follows:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:
    a. Building Type: Small Office
    b. Template: 90.1-2004
    c. Climate Zone = 5A

    Results were compared to those generated from RefBldgSmallOfficeNew2004_Chicago.idf file, which is included with EnergyPlus v8.9.0 distribution

2) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:
    a. Building Type: Large Hotel
    b. Template: 90.1-2004
    c. Climate Zone = 5A

    No comparative results were available for this test model. Output variable results were reviewed using the DVIEW software tool for reasonableness.

Required Unit Tests

None.

M2 - Example 2. Traditional Setpoint and Availability Managers

Equivalent EnergyPlus v8.9.0 .idf file

"EMSReplaceTraditionalManagers_LargeOffice"

Description

This measure demonstrates how to use OpenStudio EMS objects to model supervisory control of HVAC systems. The functionality of three traditional HVAC system managers (scheduled setpoints, mixed air setpoints, and night cycle availability) are replaced with equivalent OpenStudio EMS objects

Modeler Description

This measure replicates the functionality described in the EnergyPlus Energy Management System Application Guide, Example 2., based on user input.

Measure Type

This measure is created as an OpenStudio Measure tagged as "HVAC : HVAC Controls".

Arguments

1) AirLoopHVAC object utilizing a qualifying VAV supply fan object of Type = FanVariableVolume

   NOTE: *A fan object is a proxy for the insertion of a mixed-air setpoint manager object on the inlet node of the fan object (if one is not already present) by the Openstudio forward translator at OpenStudio runtime. The setpoint manager object inserted by the forward translator will, under this qualifying condition,* ***always*** *be of a class "SetpointManagerMixedAir". This measure will preemptively place a SetpointManagerMixedAir object on the inlet of the fan object to work correctly.*

   a. Type = Choice List (Populate with qualified objects, include additional choice "None")
   b. Default Value = None
   c. Required = True

2) AirLoopHVAC object utilizing a qualifying Availability Manager object of type = AvailabilityManager:NightCycle

3) EMS Reporting:
    a. InternalVariableAvailabilityDictionaryReporting
        i. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
    b. EMSRuntimeLanguageDebugOutputLevel
        i. Choice List for level of EMS Reporting: (““None”“, ““ErrorsOnly”“, ““Verbose”“)
    c. ActuatorAvailabilityDictionaryReporting value
        i. Choice List for level of EMS Reporting: (““None”“, ““NotByUniqueNames”“, ““Verbose”“)

Error / Warning / NA Messages

1) An N/A message will be generated if the model has zero qualified AirLoopHVAC objects

Measure Pseudocode

1) Collect measure argument for the AirLoopHVAC and EMS Reporting levels
2) Create new Openstudio schedule object for Seasonal-Reset-Supply-Air-Temp-Sch
3) Retrieve outlet nodes of AirLoop Outdoor Air Mixer, Cooling Coil and Heating Coil
4) Loop through Thermal Zones connected to AirLoopHVAC
    a. For each zone, create new EMS Sensor object representing the Zone Mean Air temp
    b. Retrieve htg and clg T-stat objects
        i. This measure will use the htg and clg T-stat from the last zone in the loop
5) Create new EMS sensor objects
    a. Schedule value of Thermal Zone Heating Setpoint
    b. Schedule value of Thermal Zone Cooling Setpoint
    c. Schedule value of the Seasonal Reset Supply Air Temp Schedule
    d. AirLoopHVAC VAV Fan Outlet Node Temperature
    e. AirLoopHVAC VAV Fan Inlet Node Temperature

6) Create new EMS Actuator objects
    a. AirLopHVAC Availability Status
    b. AirLoopHVAC Supply Outlet Node Temperature setpoint
    c. AirLoopHVAC Outdoor Air Mixer Outlet Node Temperature Setpoint
    d. Cooling Coil Outlet Node Temperature Setpoint
    e. Heating Coil Outlet Node Temperature Setpoint

7) Create new EMS Program Objects

a. Program for setting the Discharge Temperature Setpoint on the AirLoopHVAC Supply Outlet Node
b. Program object for managing the mixed air setpoint managers to account for injection of fan heat
c. Program object for setting AirLopHVAC Availability Status to manage AirLoopHVAC Night Cycle Availability based on a 0.833 Deg C offset

8) Create EMS Program Calling Manager object

9) Configure OutputEnergyManagementSystem object

Measure Testing

Exercise the completed measure against these test models, using the “Always Run” measure application method:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:

   a. Building Type: Large Office
   b. Template: 90.1-2004
   c. Climate Zone = 5A

   Compare results to those generated from RefBldgLargeOfficeNew2004_Chicago.idf including EMS code which is included with EnergyPlus v8.9.0 distribution.

2) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:

   a. Building Type: Large Hotel
   b. Template: 90.1-2004
   c. Climate Zone = 5A

   No comparative results are available for this test model. View output variable results using DVIEW for reasonableness.

Required Unit Tests

None.

M3 Example 3. Hygro-thermal Window Opening Control for Airflow Network

Note: This example EnergyPlus EMS measure will not be duplicated in OpenStudio due to deficiencies in the OpenStudio SDK/API that prevent using OpenStudio to access required model objects. EnergyPlus v8.9.0 does include an example .idf file, “EMSAirflowNetworkOpeningControlByHumidity.idf” which demonstrates EMS features. However, many of the Natural Ventilation and Duct Leakage objects used by this file were not supported by OpenStudio v2.4.0 *(when the project work scope was generated). While Airflow Network Objects are supported in the API documentation for OpenStudio v2.6.0, the project team agreed that this work scope would not include replicating EMS Example #3.

Description

N/A

Modeler Description

N/A

Measure Type

N/A

Arguments

N/A

Error / Warning / NA Messages

N/A

Measure Pseudocode

N/A

Measure Testing

N/A

Required Unit Tests

N/A

M4: Example 4. Halt Program Based on Constraint

Equivalent EnergyPlus v8.9.0 .idf file

“EMSTestMathAndKill”

Description

This measure replicates the functionality described in the EnergyPlus Energy Management System Application Guide, Example 4., based on user input.

Modeler Description

This measure demonstrates how results from custom and intermediate calculations evaluated during an OpenStudio simulation workflow can be used to trigger a ““graceful failure”“ based on a predetermined limit or constraint. Exercising this feature can be useful when users are attempting to efficiently manage resources needed for a simulation study involving a large parameter space. When used properly, this feature effectively provides dynamic ““pruning”“ of the overall solution space based on user-defined triggers for halting a simulation.

In addition to selecting a thermal zone, users will be asked to define both lower (a.bc) and upper (x.yz) PMV limits for triggering the ““graceful failure”“. An EMS Trend Variable will be configured for the Thermal Zone output variable named “Zone Thermal Comfort Fanger Model PMV”. If the trended average of the output variable is greater that a user-defined value (a.bc), the measure will fail and generate an error code of “900a.bc”. If a trended average the output variable greater than the user-defined value (x.yz), the measure will gracefully fail with the error code 100x.yz.

Measure Type

This measure was created as an OpenStudio Measure tagged as “HVAC : HVAC Controls”.

Measure Arguments

1) Qualifying Thermal Zone Object (Only consider conditioned Thermal Zone objects)

    a. Type = Choice List (Populate with qualified objects, include additional choice “None”)
    b. Default Value = None
    c. Required = True

2) ASHRAE 55 Comfort Warning

a. Type = Boolean
b. Required = True
c. Default Value = false

3) Mean Radiant Temperature Calculation Type
   a. Type = Choice List: ”ZoneAverage”, ”SurfaceWeighted”
   b. Required = True
   c. Default Value = “ZoneAveraged”

4) Qualifying Work Efficiency Schedule (Only consider schedules where ScheduleTypeLimits = “Fractional”)

   a. Type = Choice List (Populate with qualified objects, include additional choice “None”)
   b. Default Value = None
   c. Required = True

5) Qualifying Clothing Insulation Schedule (Only consider schedules where ScheduleTypeLimits = “ClothingInsulation”)

   a. Type = Choice List (Populate with qualified objects, include additional choice “None”)
   b. Default Value = 1st object in the qualifying schedule object array
   c. Required = True

6) Qualifying Air Velocity Schedule (Only consider schedules where ScheduleTypeLimits = “Velocity”)

   a. Type = Choice List (Populate with qualified objects, include additional choice “None”)
   b. Default Value = None
   c. Required = True

7) Model Timestep

   a. Type = Choice List (“60”, “30”, “15”, “10”, “6”)
   b. Default Value = “15”
   c. Required = True

8) Length of time to trend PMV (hours)

a. Type = Choice List (“2”, “3”)
b. Default Value = “2”
c. Required = True

9) Minimum Thermal Zone PMV Value to Trigger ““graceful failure”“
a. Type = Double Precision
b. Min Allowable Error Value < = -3.0
c. Max Allowable Error Value > = 3.0
d. Min Allowable Warning Value < = N/A
e. Max Allowable Warning Value > = N/A
f. Required = True
g. Default Value = 1.3

10) Maximum Zone PMV Value to Trigger ““graceful failure”“
a. Type = Double Precision
b. Min Allowable Error Value < = -3.0
c. Max Allowable Error Value > = 3.0
d. Min Allowable Warning Value < = N/A
e. Max Allowable Warning Value > = N/A
f. Required = True
g. Default Value = 2.5

11) EMS Reporting:
a. InternalVariableAvailabilityDictionaryReporting
i. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
b. EMSRuntimeLanguageDebugOutputLevel
i. Choice List for level of EMS Reporting: (““None”“, ““ErrorsOnly”“, ““Verbose”“)
c. ActuatorAvailabilityDictionaryReporting value
i. Choice List for level of EMS Reporting: (““None”“, ““NotByUniqueNames”“, ““Verbose”“)

Error / Warning / NA Messages

Generate N/A message if model has zero conditioned thermal zones

Generate Error Message if Maximum Zone PMV Value < Minimum Zone PMV Value

Generate Error Message if Maximum Zone PMV Value == Minimum Zone PMV Value

Measure Pseudocode

1) Gather measure arguments
2) Check measure arguments, write Error and N/A messages as applicable.
3) Write Initial Conditions Message
4) Loop through user-selected thermal zone(s)
    a. Loop through all space(s) connected to the thermal zone
        i. If there is an OS SpaceType already associated with the space. For each people object associated with the spacetype
            1. Check to see if the People object already has a Work Efficiency Schedule.
                a. Replace the existing Work Efficiency Schedule if there was a Work Efficiency Schedule assigned previously.
                b. Assign a new Work Efficiency Schedule if there was no Work Efficiency Schedule assigned previously.
            2. Check to see if the People object already has a Clothing Insulation Schedule.
                a. Replace the existing Clothing Insulation Schedule if there was a Clothing Insulation Schedule assigned previously.
                b. Assign a new Clothing Insulation Schedule if there was no Clothing Insulation Schedule assigned previously.
            3. Check to see if the People object already has an Air Velocity Schedule:
                a. Replace the existing Air Velocity Schedule if there was an Air Velocity Schedule assigned previously.
                b. Assign a new Air Velocity Schedule if there was no Air Velocity Schedule assigned previously
            4. For the People Definition object associated with each People object:
                a. Check if ASHRAE 55 Thermal Comfort warnings have already been set. Set them as needed.
                b. Check Mean Radiant Temperature calculation setting. Set to user-argument value if necessary.
                c. Set E+ Thermal Comfort Model Type = “Fanger”
            5. Apply steps (1 – 4) above for all people objects that are ““hard assigned”“ to the space.
    b. Create new EnergyManagementSystem:Sensor object.for measuring the Zone Fanger Model PMV value.
    c. Create new EnergyManagementSystem:TrendVariable object for trending the PMV sensor value over time, per user-arguments.

   d. Create new EnergyManagementSystem:GlobalVariable object to hold the running trend value average for necessary calculations.
   e. Create new EnergyManagementSystem:Subroutine object with logic for comparing the running average zone PMV and comparing it to limits set by user arguments. If PMV trend limits are exceeded, halt the simulation and generate the appropriate error code.
   f. Create new EnergyManagementSystem:Program object determine running PMV average and call subroutine
   g. Create new EnergyManagementSystem:ProgramCallingManager object and configure to run the EMS program at the end of the zone timestep but before zone reporting.
   h. Create new EnergyManagementSystem:OutputVariable object and map it to the Global Variable representing the Zone PMV trend variable
   i. Create new E+ Output variable and map to the EMS Trend Variable
   j. Write Info messages based on user arguments
5) Write Final Conditions Message

Measure Testing

Exercise the completed measure against these test models, using the “Always Run” measure application method:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:
   a. Building Type: Small Hotel
   b. Template: 90.1-2004
   c. Climate Zone = 5A (Chicago)

   Compare results to those generated from RefBldgSmallHotelNew2004_Chicago.idf example including EMS code which is included with EnergyPlus v8.8 distribution

2) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:
   a. Building Type: Large Hotel
   b. Template: 90.1-2004
   c. Climate Zone = 5A

Required Unit Tests

None

M5: Example 5. Computed Schedule

Equivalent EnergyPlus v8.9.0 .idf file

“EMSCustomSchedule.idf”

Description

This measure replicates the functionality described in the EnergyPlus v8.9.0 Energy Management System Application Guide, Example 5., based on user input.

Modeler Description

The example demonstrates the use of a thermostat schedule object as and EMS actuator object. The EMS program alters the scheduled values as a function both of hour of day and day of week.

Measure Type

This measure will be created as an OpenStudio Measure tagged as “HVAC : HVAC Controls”.

Measure Arguments

1) Qualifying Thermal Zone Object (Only consider conditioned Thermal Zone objects)

    a. Type = Choice List (Populate with qualified objects, include additional choice “All”)
    b. Default Value = None
    c. Required = True

2) EMS Reporting:
    a. InternalVariableAvailabilityDictionaryReporting
        ii. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
    b. EMSRuntimeLanguageDebugOutputLevel
        iii. Choice List for level of EMS Reporting: (““None”“, ““ErrorsOnly”“, ““Verbose”“)
    c. ActuatorAvailabilityDictionaryReporting value
        iv. Choice List for level of EMS Reporting: (““None”“, ““NotByUniqueNames”“, ““Verbose”“)

Error / Warning / NA Messages

Generate N/A message if model has zero conditioned thermal zones

Measure Pseudocode

1) Gather measure arguments
2) Write Initial and Final Conditions Messages
3) Write Error and N/A message(s)

4) Create new OS:Schedule:Constant objects for heating (21C) and cooling (24C) setpoints
5) For each user-selected conditioned Thermal Zone(s)
    a. Replace existing clg thermostat schedule with OS:Schedule:Constant schedule object
    b. Replace existing htg thermostat schedule with OS:Schedule:Constant schedule object
6) Create EMS Actuator Object representing htg T-stat schedule created in (3) above
7) Create EMS Actuator Object representing clg T-stat schedule created in (3) above
8) Create EnergyManagementSystem:Program object and configure with logic for setting htg t-stat values based on Month, DayOfMonth, Holiday, DayofWeekand Hour
9) Create EnergyManagementSystem:Program object and configure with logic for setting clg t-stat values based on Month, DayOfMonth, Holiday, DayofWeekand Hour
10) Create and configure EnergyManagementSystem:ProgramCallingManager object to call htg and clg programs at "BeginTimeStepBeforePredictor" calling point
11) Write Info Messages
12) Write Final Conditions Message

Measure Testing

Exercise the completed measure against these test models, using the "Always Run" measure application method:

3) DOE Reference Building generated via "Create DOE Prototype Building" OpenStudio Measure. Use arguments:

    a. Building Type: Small Office
    b. Template: 90.1-2004
    c. Climate Zone = 5A

    Compare results to those generated from RefBldgSmallOfficeNew2004_Chicago.idf including EMS code, which is included with EnergyPlus v8.8 distribution

4) DOE Reference Building generated via "Create DOE Prototype Building" OpenStudio Measure. Use arguments:

    a. Building Type: Large Hotel
    b. Template: 90.1-2004
    c. Climate Zone = 5A

    Ensure the model runs correctly when this measure is applied.

Required Unit Tests

None

M6: Example 6. Window Shade Control

Note: This Example EnergyPlus EMS measure was not duplicated as a part of this project due to deficiencies in the OpenStudio SDK/API that prevent using OpenStudio to access required model objects. Specifically, the WindowProperty:ShadeControl Object is not supported in the API documentation for OpenStudio v2.6.0. This would require the WindowProperty:ShadeControl Object to be accessed through an EnergyPlus measure. While developing an OpenStudio EMS measures to integrate with a follow-on EnergyPlus measures is possible, it is a fragile process that can be easily broken by other OpenStudio measures and unanticipated user actions, at best.

Description

N/A

Modeler Description

N/A

Measure Type

N/A

Arguments

N/A

Error / Warning / NA Messages

N/A

Measure Pseudocode

N/A

Measure Testing

N/A

Required Unit Tests

N/A

M7: Example 7. Constant Volume Purchased Air System

Equivalent EnergyPlus v8.9.0 .idf file

"EMSConstantVolumePurchasedAir.idf"

Description

This measure replicates the EMS functionality described in Example 7 from the EnergyPlus V8.9 EMS Application Guide.

Modeler Description

This measure asks the user which existing conditioned thermal zones to convert to be served by an Autosized ZoneHVACIdealLoadsAirSystems from a choice list. The choice list will only be populated by Thermal zones which are (1) conditioned and (2) served only by ZoneHVAC Equipment objects, which this measure will delete. The measure configures the ZoneHVACIdealLoadsAirSystems with user-defined values for the supply airflow rates (cfm/ft2), leaving air temperature (Deg F) and leaving air humidity ratios (lb H2O / lb dry air) for both cooling and heating modes.

Measure Type

This measure will be created as an OpenStudio Measure tagged as "HVAC : HVAC Controls".

Measure Arguments

1) Qualifying Thermal Zone Object (Only consider conditioned Thermal Zone objects served by i ZoneHVACIdealLoadsAirSystem objects)

    a. Type = Choice List (Populate with qualified objects, include additional choice "None")
    b. Default Value = None
    c. Required = True

2) Fixed Cooling Supply Airflow (cfm/sqft) to be delivered to the selected thermal zone
    a. Type = Double Precision
    b. Min Allowable Error Value < = 0
    c. Max Allowable Error Value > = 3.0
    d. Min Allowable Warning Value < = N/A
    e. Max Allowable Warning Value > = N/A
    f. Default Value = 1.0 (cfm/sqft)

3) Fixed Heating Supply Airflow (cfm/sqft) to be delivered to the selected thermal zone
    a. Type = Double Precision
    b. Min Allowable Error Value < = 0
    c. Max Allowable Error Value > = 3.0
    d. Min Allowable Warning Value < = N/A
    e. Max Allowable Warning Value > = N/A
    f. Default Value = 1.0 (cfm/sqft)
4) Fixed Cooling Supply Leaving Air Temperature (°F) to be delivered to the selected thermal zone
    a. Type = Double Precision
    b. Min Allowable Error Value < = 42
    c. Max Allowable Error Value > = 65
    d. Min Allowable Warning Value < = N/A
    e. Max Allowable Warning Value > = N/A
    f. Default Value = 55

5) Fixed Heating Supply Leaving Air Temperature (°F) to be delivered to the selected thermal zone
    a. Type = Double Precision
    b. Min Allowable Error Value < = 90
    c. Max Allowable Error Value > = 120
    d. Min Allowable Warning Value < = N/A
    e. Max Allowable Warning Value > = N/A
    f. Default Value = 105

6) Fixed Cooling Supply Leaving Air Humidity Ratio (lb H2O / lb Dry Air) to be delivered to the selected thermal zone
    a. Type = Double Precision
    b. Min Allowable Error Value < = 0.006
    c. Max Allowable Error Value > = 0.017
    d. Min Allowable Warning Value < = N/A
    e. Max Allowable Warning Value > = N/A
    f. Default Value = 0.009

7) Fixed Heating Supply Leaving Air Humidity Ratio (lb H2O / lb Dry Air) to be delivered to the selected thermal zone
    a. Type = Double Precision
    b. Min Allowable Error Value < = 0.006

c. Max Allowable Error Value > = 0.017
d. Min Allowable Warning Value < = N/A
e. Max Allowable Warning Value > = N/A
f. Default Value = 0.015

8) EMS Reporting:
    d. InternalVariableAvailabilityDictionaryReporting
        v. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
    e. EMSRuntimeLanguageDebugOutputLevel
        vi. Choice List for level of EMS Reporting: (““None”“, ““ErrorsOnly”“, ““Verbose”“)
    f. ActuatorAvailabilityDictionaryReporting value
        vii. Choice List for level of EMS Reporting: (““None”“, ““NotByUniqueNames”“, ““Verbose”“)

Error / Warning / NA Messages

Generate N/A message if model has zero conditioned thermal zones served by an ZoneHVACIdealLoadsAirSystem object

Generate N/A message if model has VRF outdoor unit HVAC systems

Generate Error Messages if measure arguments exceed thresholds

Measure Pseudocode

1) Gather measure arguments
2) Generate Error and N/A messages
3) Generate Initial Conditions Message
4) For each of the user-selected Thermal Zone(s)
    a. Remove currently attached ZoneHVACEquipment objects
    b. Delete orphaned plant loops (those having no demand side components) due to a) above
    c. Create a new ZoneHVACIdealAirLoadsHVACSystem Object
        i. Configure object with user-defined arguments
    d. Add ZoneHVACIdealAirLoadsHVACSystem Object to Thermal Zone
    e. Create EnergyManagementSystem:ProgramCallingManager object
    f. Create new EnergyManagementSystem:Program object for determining “Purchased Air State” and stub it out
    g. Create new EnergyManagementSystem:Program object for setting “Purchased Air State” and stub it out

h. Create EnergyManagementSystem:GlobalVariable object for holding current Thermal Zone ““state”“
i. Create EnergyManagementSystem:Sensor object for holding current Thermal Zone ““Zone Predicted Sensible Load to Setpoint Heat Transfer Rate”“
j. Create EnergyManagementSystem:Actuator object for EMS Actuator Objects for setting the Thermal Zone Ideal Loads Air System Air Mass Flow Rate, Supply Air Temp and Supply Air Humidity Ratio
    i. For each Thermal Zones selected by the user, append EMS Logic blocks for determining the ““purchased air state”“ and setting the ideal air loads HVAC system leaving air properties to the (2) Program objects created above

5) Generate Final Conditions message

Measure Testing

Exercise the completed measure against these test models, using the “Always Run” measure application method:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:

    a. Building Type: Small Hotel
    b. Template: 90.1-2004
    c. Climate Zone = 5A (Chicago)

    Compare results to those generated from RefBldgSmallHotelNew2004_Chicago.idf example including EMS code which is included with EnergyPlus v8.9 distribution

Required Unit Tests

None

M8: Example 8. System Sizing with Discrete Package Sizes

Equivalent EnergyPlus v8.9.0 .idf file

"EMSDiscreteAirSystemSizes.idf"

Description

This measure will replicate the functionality described in the EnergyPlus v 8.9.0 Energy Management System Application Guide, Example 8, based on user input.

Modeler Description

This measure demonstrates how EMS functions can be used to demonstrate how information from a sizing run can be used to select HVAC Equipment from nominal product sizes where unit total capacity is directly related to the unit supply airflow (1 ton = 1200 cfm, 1.5 ton = 1600 cfm, etc.) of commercial packaged single-zone HVAC air systems.

This measure is designed to work on AirLoops with packaged DX cooling equipment only. EMS functions will be used to extract the design supply airflow generated from system auto-sizing calculations. An interval variable is used to override the Sizing:System - ""Intermediate Air System Main Supply Volume Flow Rate"" value variable

This measure approximates the manner that appropriate ""real world"" equipment selections are made by HVAC design engineers. The table below will be used to map to the Maximum Flow rate of the Packaged Unit Fan:ConstantVolume object.

Threshold Sizing Supply Airflow (cfm)	Selection Supply Airflow (cfm)
$0 < V \leq 1200$	$V = 1200$
$1200 < V \leq 1600$	$V = 1600$
$1600 < V \leq 2000$	$V = 2000$
$2000 < V \leq 2360$	$V = 2360$
$2360 < V \leq 3000$	$V = 3000$
$3000 < V \leq 3400$	$V = 3400$
$3400 < V \leq 4000$	$V = 4000$

NOTE: If an Autosized value for V is greater then 4,000 cfm, the Openstudio EMS program will fail, and write a severe error into the eplusout.err file with an error code value of 666.

Measure Type

This measure will be created as an OpenStudio Measure tagged as “HVAC : HVAC Controls”.

Measure Arguments

1) Qualifying AirLoopHVAC Object (Only consider AirLoopHVAC objects having conditioned Thermal Zone objects having a Supply Fan of type = FanConstantVolume objects

    a. Type = Choice List (Populate with qualified objects, include additional choice “None”)
    b. Default Value = None
    c. Required = True

2) EMS Reporting:
    a. InternalVariableAvailabilityDictionaryReporting
        i. Choice List: ““None””, ““NotByUniqueNames””, ““Verbose””
    b. EMSRuntimeLanguageDebugOutputLevel
        ii. Choice List for level of EMS Reporting: (““None””, ““ErrorsOnly””, ““Verbose””)
    c. ActuatorAvailabilityDictionaryReporting value
        iii. Choice List for level of EMS Reporting: (““None””, ““NotByUniqueNames””, ““Verbose””)

Error / Warning / NA Messages

Generate N/A message if model has qualifying AirLoopHVAC objects

Measure Pseudocode

1) Gather measure arguments
2) Generate Initial Conditions Message
3) Generate Error and N/A messages
4) Stub out a new EnergyManagementSystem:Program object for resizing a PSZ AirLoop
5) Create a new EnergyManagementSystem:ProgramCallingManager object to execute at the “EndOfSystemSizing” calling point
6) Create a new EnergyManagementSystem:GlobalVariable object and configure to hold a discrete value mapped to argDiscreteMainVdot
7) Create a new EnergyManagementSystem:GlobalVariable object and configure to hold the value mapped to argMainVdot

8) Create a new EnergyManagementSystem:SubroutineProgram object and configure to set supply airflow based on multiple logic statements describing how to apply discrete supply airflow rates based minimum and maximum supply airflow rates from system autosized values
9) Loop through all AirLoopHVAC objects matching user arguments
    a. Create a new EnergyManagementSystem:InternalVariable object and configure to determine Intermediate Air System Main Supply Volume Flow Rate
    b. Create a new EnergyManagementSystem:Actuator object and configure to use for holding Main Supply Volume Flow Rate from the system sizing object of the user selected system
    c. Append new logic to EnergyManagementSystem:Program stub created in (4) above
10) Configure EMS OutputEnergyManagementSystem object per user arguments
11) Generate Final Conditions Message

Measure Testing

Exercise the completed measure against these test models, using the “Always Run” measure application method:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:

    a. Building Type: Strip Mall
    b. Template: 90.1-2004
    c. Climate Zone = 5A (Chicago)

    Compare results to those generated from “RefBldgStripMallNew2004_Chicago.idf.” example including EMS code which is included with EnergyPlus v8.9 distribution.

    NOTE: If the user selects an argument of “All AirLoops”, there will be autosized values that exceed the 4,000 cfm limit and that will halt the EMS Program via a severe error with an error code value of 666.

Required Unit Tests

None

M9: Example 9. Demand Management

Equivalent EnergyPlus v8.9.0 .idf file

“EMSDemandManager_LargeOffice.idf”

Description

This measure will replicate the functionality described in the EnergyPlus v8.9.0 Energy Management System Application Guide, Example 9, based on user input.

Modeler Description

This measure will demonstrate how EMS functions can be used to apply controls to a large office model which mimics common ““peak shaving”“ algorithms available in building energy management and control systems (EMCS). The example measure will impose limits to the whole building peak electrical power draw by imposing one of four different peak demand control states (logic conditions) based on comparing (current power levels) to (“user defined” target levels). The example program introduces a trend variable of control modes to represent a numerical dampening effect and limit control mode ““bouncing”“. The four control models are hard-wired to implement these levels of reductions to zone lighting power and zone cooling thermostats

Control State	Lighting Power Adjustment Factor	Cooling Thermostat Offset
0	None	None
1	0.9	+1.44ºF (+0.8ºC)
2	0.8	+2.70ºF (+1.5ºC)
3	0.7	+3.60ºF (+2.0ºC)

For this example, user argument will be exposed to define the monthly Peak kW Demand (kW target) values. The default values for these arguments represent a user defined % reduction of the monthly peak kW values derived from the “seed”.osm model, when run ***without*** enabling peak shaving algorithms.

The monthly peak demand value used in the EMS program is mapped to the PeakDemand column of the E+ “Native Variables” table, generated as part of standard E+ output data.

Measure Type

This measure will be created as an OpenStudio Measure tagged as “HVAC : HVAC Controls”.

Measure Arguments

1) Qualifying Lighting Control Fractional Schedule (Only consider fractional schedules attached to lighting loads attached to model spaces)
    a. Type = Choice List (Populate with qualified objects)
    b. Default Value = None
    c. Required = True
2) Qualifying Thermostat Temperature Schedule (Measure will only alter the cooling t-stat temperature for attached thermal zones)
    a. Type = Choice List (Populate with qualified objects)
    b. Default Value = None
    c. Required = True
3) Monthly Electricity Demand Reduction Target (%)
    a. Type = Double Precision
    b. Min Allowable Error Value < = 0.00
    c. Max Allowable Error Value > = 0.35
    d. Min Allowable Warning Value < = N/A
    e. Max Allowable Warning Value > = N/A
    f. Default Value = 15

Error / Warning / NA Messages

Generate Error messages if user arguments are not found in the model

Measure Pseudocode

1) Gather measure arguments
2) Write Initial Conditions message
3) Error and N/A messages
4) Hard code monthly peak demand values from previous model run into array.
5) Create new EnergyManagementSystem:Global Variables for:
    a. Target Demend Level
    b. Current Demand Manager State (0, 1, 2,or 3)
    c. Current Demand Level
    d. Current Demand Trend Direction
6) Create a new EnergyManagementSystem:Actuator object for changing the value of the cooling setpoint schedule
7) Create new EnergyManagementSystem:Actuator object(s) for changing the thermal zone lighting electric power level value

   a. Create new EnergyManagementSystem:InternalVariable object(s) for holding the lighting power design level used in subsequent EMS program statements
8) Create a new EnergyManagementSystem:Sensor object representing the current Whole Building, Facility Total Electric Demand Power
9) Create a new EnergyManagementSystem:Sensor object representing the current value of the cooling setpoint schedule
10) Create a new EnergyManagementSystem:Sensor object representing the current value of the lighting setpoint schedule
11) Create a new EnergyManagementSystem:TrendVariable object and configure to evaluate the Whole Building Facility Total Electric Demand Power sensor over 144 timesteps
12) Create a new EnergyManagementSystem:TrendVariable object and configure to evaluate the demand manager state sensor over 48 timesteps
13) Create a new EnergyManagementSystem:SubroutineProgram object and configure to remove the demand base controls (current demand manager state == 0)
14) Create new EnergyManagementSystem:SubroutineProgram objects and configure to determine the current demand manager state to (0, 1, 2, or 3) based on trend variable results and other logic
15) Create a new EnergyManagementSystem:SubroutineProgram object and configure to set the demand manager state to (0, 1, 2, or 3) based on trend variable results and other logic
16) Create EnergyManagementSystem:Program object and configure to call subroutine objects to dispatch controls changes based on current value of demand manager state (0, 1, 2, 3)
17) Create EnergyManagementSystem:Program object and configure to call subroutine objects to determine the current state of the demand manager
18) Create a new EnergyManagementSystem:ProgramCallingManager object configured to call EMS Programs at the “BeginTimeStepBeforePredictor” calling point
19) Configure OutputEnergyManagementSystem properties with user arguments
20) Generate Final Conditions Message

Measure Testing

Exercise the completed measure against these test models, using the “Always Run” measure application method:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:

   a. Building Type: Large Office
   b. Template: 90.1-2010
   c. Climate Zone = 5A (Chicago)

   Compare results to those generated from “EMSDemandManager_LargeOffice.idf.” example including EMS code which is included with EnergyPlus v8.9 distribution

Required Unit Tests

None

M10: Example 10. Plant Loop Override Control

Equivalent EnergyPlus v8.9.0 .idf file

“EMSPlantLoopOverrideControl.idf”

Description

This measure will replicate the functionality described in the EnergyPlus v 8.9.0 Energy Management System Application Guide, Example 10, based on user input.

Modeler Description

This measure will demonstrate how common warnings generated in .err files can be ““trapped”“ and eliminated by applying conditional logic in an OpenStudio EMS program. In this case, the warning we want to eliminate relates to improper cooling tower operation, and the warning states that the tower temperature is operating below a low temperature limit. The measure will use an EMS program that checks to see if the outdoor temperature is below the allowable temperature for a cooing tower to operate. If this is true, the operation of the plant loop belonging to the cooling tower will be disabled. To properly disable the plant loop, both the loop equipment flow control objects (the condenser pump) and the parent loop itself will be disabled. The loop pump will be disabled by setting an EMS Actuator variable representing the pump mass flow rate to a value of zero. The plant loop object will be disabled by setting the On/Off supervisory control to a value of ““Off””.

Measure Type

This measure will be created as an OpenStudio Measure tagged as “HVAC : HVAC Controls”.

Measure Arguments

1) Qualifying Plant Loop Object (Only consider Plant Loop objects where Sizing Plant Loop Type = “Condenser”

    a. Type = Choice List (Populate with qualified objects, include additional choice “None”)
    b. Default Value = None
    c. Required = True

2) OA DB Temp (°F) below which to disable the Condenser Plant Loop

    a. Type = Double Precision
    b. Default Value = 42.8
    c. Minimum Warning Value = 38
    d. Maximum Warning Value = 50
    e. Required = True

3) EMS Reporting:
    a. InternalVariableAvailabilityDictionaryReporting
        i. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
    b. EMSRuntimeLanguageDebugOutputLevel
        ii. Choice List for level of EMS Reporting: (““None”“, ““ErrorsOnly”“, ““Verbose”“)
    c. ActuatorAvailabilityDictionaryReporting value
        iii. Choice List for level of EMS Reporting: (““None”“, ““NotByUniqueNames”“, ““Verbose”“)

Error / Warning / NA Messages

Generate N/A message if model has no qualifying Plant Loop objects

Measure Pseudocode

1) Gather measure arguments
2) Generate Initial Conditions message
3) Generate Error and N/A messages
4) Create new EnergyManagementSystem:Sensor object for measuring Site Outdoor Air Dry Bulb Temperature
5) Create a new EnergyManagementSystem:ProgramCallingManager object and configure it to call the program at the “InsideHVACSystemIterationLoop” Calling Point
6) Stub out a new EnergyManagementSystem:Program object
7) Loop through all user-selected Condenser Plant Loop objects
    a. Retrieve the constant or variable speed pump object associated with the supply side
    b. Create new EnergyManagementSystem:GlobalVariable object and configure for reporting status of pump flow override.
    c. Create EnergyManagementSystem:Actuator object and configure to override the condenser plant loop on/off supervisory control action.
    d. Create EnergyManagementSystem:Actuator object and configure to override condenser pump mass flow rate
    e. Create EnergyManagementSystem:OutputVariable object and configure for reporting status of pump flow override across the system timestep
    f. Append logic to the EnergyManagementSystem:Program object to set the loop status based on the outdoor air temperature sensor. The logic will set pump flow and loop control overrides.
    g. Create E+ :OutputVariable object and configure for reporting status of pump flow override across system timestep

h. Create EnergyManagementSystem:OutputVariable object and configure for reporting status of condenser loop flow override at hourly timestep
i. Create EnergyManagementSystem:ProgramCallingManager object
j. Create EnergyManagementSystem:Program object and

8) Configure EMS OutputEnergyManagementSystem object per user arguments
9) Write Final Conditions Message

Measure Testing

Exercise the completed measure against these test models, using the “Always Run” measure application method:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:

a. Building Type: Large Office
b. Template: 90.1-2004
c. Climate Zone = 5A (Chicago)

Required Unit Tests

None

M11: Example 11. Performance Curve Result Override

Equivalent EnergyPlus v8.9.0 .idf file

"EMSCurveOverride_PackagedTerminalHeatPump.idf"

Description

This measure will replicate the functionality described in the EnergyPlus v8.9.0 Energy Management System Application Guide, Example 11, based on user input.

Modeler Description

This measure will demonstrate how an OpenStudio measure calling EMS functions can be used to model the performance of HVAC equipment that cannot be represented well by using single "standard" performance curve objects (cubic, quadratic, biquadratic, etc.) For example, properly characterizing some HVAC equipment objects requires using different performance curves that cover operation of different parts of the performance regime. This measure will alter (overwrite) the Coil Cooling DX Single Speed Cooling Capacity as a function of temperature performance curve object and attributes used by the simulation if the outdoor air temperature falls below a user defined threshold.

This measure allows the user to define the biquadratic curve coefficients associated with the Coil Cooling DX Single Speed Cooling Capacity.

Measure Type

This measure will be created as an OpenStudio Measure tagged as "HVAC : HVAC Controls".

Measure Arguments

1) Qualifying Single Speed DX Cooling Coil Object (Only consider DX Cooling Coils that are children of a ZoneHVAC Equpment objects of type = PTHP)

    a. Type = Choice List (Populate with qualified objects, include additional choice "All")
    b. Default Value = All
    c. Required = True

2) OAT DB Threshold for changing Single Speed DX Cooling Coil HPACCoolCapFT curve attributes
    a. Type = Double Precision
    b. Min Allowable Error Value < = -10
    c. Max Allowable Error Value > = 130
    d. Min Allowable Warning Value < = N/A
    e. Max Allowable Warning > = N/A
    f. Default Value = 87.8 (°F)

3) Replacement HPACCoolCapFT biquadratic curve attribute - value used for A

a. Type = Double Precision
b. Min Allowable Error Value < = N/A
c. Max Allowable Error Value > = N/A
d. Min Allowable Warning Value < = N/A
e. Max Allowable Warning > = N/A
f. Default Value = 0.942567793

4) Replacement HPACCoolCapFT biquadratic curve attribute - value used for B
   a. Type = Double Precision
   b. Min Allowable Error Value < = N/A
   c. Max Allowable Error Value > = N/A
   d. Min Allowable Warning Value < = N/A
   e. Max Allowable Warning > = N/A
   f. Default Value = 0.009543347

5) Replacement HPACCoolCapFT biquadratic curve attribute - value used for C
   a. Type = Double Precision
   b. Min Allowable Error Value < = N/A
   c. Max Allowable Error Value > = N/A
   d. Min Allowable Warning Value < = N/A
   e. Max Allowable Warning > = N/A
   f. Default Value = 0.00068377

6) Replacement HPACCoolCapFT biquadratic curve attribute - value used for D
   a. Type = Double Precision
   b. Min Allowable Error Value < = N/A
   c. Max Allowable Error Value > = N/A
   d. Min Allowable Warning Value < = N/A
   e. Max Allowable Warning > = N/A
   f. Default Value = 0.011042676

7) Replacement HPACCoolCapFT biquadratic curve attribute - value used for E
   a. Type = Double Precision
   b. Min Allowable Error Value < = N/A
   c. Max Allowable Error Value > = N/A
   d. Min Allowable Warning Value < = N/A
   e. Max Allowable Warning > = N/A
   f. Default Value = 0.000005249

8) Replacement HPACCoolCapFT biquadratic curve attribute - value used for F
   a. Type = Double Precision
   b. Min Allowable Error Value < = N/A
   c. Max Allowable Error Value > = N/A
   d. Min Allowable Warning Value < = N/A
   e. Max Allowable Warning > = N/A
   f. Default Value = 0.000009720

9) EMS Reporting:

a. InternalVariableAvailabilityDictionaryReporting
    i. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
b. EMSRuntimeLanguageDebugOutputLevel
    ii. Choice List for level of EMS Reporting: (““None”“, ““ErrorsOnly”“, ““Verbose”“)
c. ActuatorAvailabilityDictionaryReporting value
    iii. Choice List for level of EMS Reporting: (““None”“, ““NotByUniqueNames”“, ““Verbose”“)

Error / Warning / NA Messages

Generate N/A message if model has qualifying Single Speed DX Coil Objects.

Measure Pseudocode

1) Gather measure arguments
2) Generate Initial Conditions message
3) Generate Error and N/A messages
4) Stub a new EnergyManagementSystem:Program object for overriding DX Single Speed clg coil capacity curves
5) Create a new EnergyManagementSystem:ProgramCallingManager object and configure to call the EMS program using the “AfterPredictorbeforeHVACManager” calling point
6) Loop through user-selected all DX Single Speed Cooling Coil object(s)
    a. Retrieve parent PTHP object in order to extract the DX coil inlet node, DX Coil outlet node and OA Mixer Outlet Node
    b. Create a new energyManagementSystem:Actuator object and configure to map to Cooling Capacity as a Function of Temperature curve
    c. Create a new EnergyManagementSystem:GlobalVariable object to hold the results of evaluating the new performance curve
    d. Create a new EnergyManagementSystem:Sensor object and configure to map to the current value of the Cooling Capacity as a Function of Temperature curve object of the user selected PTHP object
    e. Create a new EnergyManagementSystem:Sensor object and configure to map to the OA Mixer Inlet Node Pressure
    f. Create a new EnergyManagementSystem:Sensor object and configure to map to the OA Mixer Inlet Node Temperature
    g. Create a new EnergyManagementSystem:Sensor object and configure to map to the DX Cooling Coil Inlet Dry Bulb Temperature

h. Create a new EnergyManagementSystem:Sensor object and configure to map to the DX Cooling Coil Inlet Humidity Ratio
i. Create a new EnergyManagementSystem:OutputVariable object and configure for holding the current value of the Cooling Capacity as a Function of Temperature curve object
j. Create a new OpenStudio:OutputVariable object and configure for holding the current value of the Cooling Capacity as a Function of Temperature curve object
k. Create a new EnergyManagementSystem:OutputVariable object and configure for holding the old value of the Cooling Capacity as a Function of Temperature curve object averaged over the zone timestep
l. Create a new OpenStuio:OutputVariable object and configure for holding the old value of the Cooling Capacity as a Function of Temperature curve object averaged over the zone timestep
m. Create a new EnergyManagementSystem:OutputVariable object and configure for holding the new value of the Cooling Capacity as a Function of Temperature curve object averaged over the zone timestep
n. Create a new OpenStudio:OutputVariable object and configure for holding the new value of the Cooling Capacity as a Function of Temperature curve object averaged over the zone timestep
o. Append logic to the EnergyManagementSystem:Program object and to set curve coefficients based on current value of outside temperature

7) Create a new OpenStudio:OutputVariable object and configure for holding the Site Outdoor Air Drybulb temperature value
8) Generate Final Conditions Message

Measure Testing

Exercise the completed measure against these test models, using the “Always Run” measure application method:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:

    a. Building Type: Large Office
    b. Template: 90.1-2004
    c. Climate Zone = 5A (Chicago)

Required Unit Tests

None

M12: Example 12. Variable Refrigerant Flow System Override

Equivalent EnergyPlus v8.9.0 .idf file

None

Description

This measure will replicate the functionality described in the EnergyPlus v 8.9.0 Energy Management System Application Guide, Example 12, based on user input.

Modeler Description

This measure will demonstrate how an OpenStudio measure calling EMS functions can be used to override specified thermostat control logic and set alternate modes of operation. This EMS measure sets a specific (user defined) indoor VRF terminal unit to operate at a specific (user-defined) part load ratio, constrained by operate minimum and maximum outdoor temperature limits of the paired condenser unit.

Measure Type

This measure will be created as an OpenStudio Measure tagged as “HVAC : HVAC Controls”.

Measure Arguments

1) Qualifying Outdoor ZoneHVAC VRF Units
    a. Type = Choice List (Populate with qualified objects, include additional choice “All”)
    b. Default Value = All
    c. Required = True
2) Fixed PLR to set for attached Indoor VRF Units
    a. Type = Double Precision
    b. Min Allowable Error Value < = 0
    c. Max Allowable Error Value > = 1
    d. Min Allowable Warning Value < = N/A
    e. Max Allowable Warning > = N/A
    f. Default Value = 0.5
3) EMS Reporting:
    a. InternalVariableAvailabilityDictionaryReporting
        iv. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
    b. EMSRuntimeLanguageDebugOutputLevel

v. Choice List for level of EMS Reporting: (““None””, ““ErrorsOnly””, ““Verbose””)

c. ActuatorAvailabilityDictionaryReporting value

vi. Choice List for level of EMS Reporting: (““None””, ““NotByUniqueNames””, ““Verbose””)

Error / Warning / NA Messages

Generate N/A message if model has qualifying Outdoor VRF Units.

Measure Pseudocode

1) Gather measure arguments
2) Generate Initial Conditions Message
3) Generate Error and N/A message(s)
4) Create a new EnergyManagementSystem:ProgramCallingManager object set to run EMS program at “BeginNewEnvironment” calling point
5) Create a new EnergyManagementSystem:ProgramCallingManager object object set to run EMS program at “InsideHVACSystemIteratonLoop” calling point
6) Sub out new EnergyManagementSystem:Program object and configure with EMS logic statements for PLR of user selected terminal unit
7) Stub out new EnergyManagementSystem:Program object and configure to set VRF Status (Off/Heat/Cool)
8) Loop through all VRF indoor Unit Zone Equipment objects to find user-selected object
    a. Create EnergyManagementSystem:Actuator object and configure to map to Operating Mode variable of the Outdoor VRF Heat Pump object associated with the user-selected VRF Indoor unit.
    b. Create EnergyManagementSystem:Actuator object and configure to map to Part Load Ratio of the user-selected VRF Indoor unit.
    c. Create EnergyManagementSystem:OutputVariable object and configure for holding the current value of the VRF Outdoor Unit Operating Mode
    d. Create several EnergyManagementSystem:GlobalVariable objects and configure for holding the VRF Status Off, VRF Status Cooling and VRF Status Heating
    e. Create several EnergyManagementSystem:OutputVariable objects to track the detailed values of the VRF Indoor Unit Control status, VRF Outdoor Unit Heat Pump Operating Mode, Heating and Cooling Coil Runtime fractions
9) Write Final Conditions Messages

Measure Testing

Exercise the completed measure against these test models, using the “Always Run” measure application method:

1) DOE Reference Building generated via “Create DOE Prototype Building” OpenStudio Measure. Use arguments:

    a. Building Type: Small Office
    b. Template: 90.1-2004
    c. Climate Zone = 5A (Chicago)

    Once generated, delete the existing HVAC system and apply an auto-sized heat pump VRF system (5 terminal unit, one outdoor unit) to this building.

Required Unit Tests

None

M13: Example 13. Surface Construction Actuator for Thermochromic Window

Equivalent EnergyPlus v8.9.0 .idf file

“EMSThermochromicWindow.idf”

Description

This measure will replicate the functionality described in the EnergyPlus v8.9.0 Energy Management System Application Guide, Example 13, based on user input.

Modeler Description

This measure will demonstrate how an OpenStudio measure calling EMS functions can be used to investigate dynamic envelope technologies such as emulating thermochromic window performance using EMS actuators and control types. This measure will replace the construction description of a user-selected window based on the outside surface temperature of that window, evaluated at each timestep.

Measure Type

This measure will be created as an OpenStudio Measure tagged as “HVAC : HVAC Controls”.

Measure Arguments

1) Fixed Window Subsurface to apply EMS Chromatic Window Properties to
    a. Type = Choice List (Populate with qualified objects, include additional choice “All”)
    b. Default Value = All
    c. Required = True
2) EMS Reporting:
    a. InternalVariableAvailabilityDictionaryReporting
        vii. Choice List: ““None”“, ““NotByUniqueNames”“, ““Verbose”“
    b. EMSRuntimeLanguageDebugOutputLevel
        viii. Choice List for level of EMS Reporting: (““None”“, ““ErrorsOnly”“, ““Verbose”“)
    c. ActuatorAvailabilityDictionaryReporting value
        ix. Choice List for level of EMS Reporting: (““None”“, ““NotByUniqueNames”“, ““Verbose”“)

Error / Warning / NA Messages

1) None

Measure Pseudocode

1) Gather user arguments
2) Generate Initial Conditions message
3) Generate Error and Warning message(s)
4) Create 21 new window construction objects representing the layered performance at different outside surface temperature conditions. Load performance data from resource .idf file titled "EMSThermochromicWindow.idf "supplied with measure
5) Create 19 new EnergyManagementSystem:ConstructionIndexVariable objects to associate with each bin value for window surface temperature and related performance
6) Create a new EnergyManagementSystem:Sensor object and configure to map to the outside face temperature of the window object that will be altered
7) Create a new EnergyManagementSystem:Actuator object to map to the construction state of the window object that will be altered
8) Create a new EnergyManagementSystem:ProgramCallingManager object set to call the EMS program the the "BeginTimestepBeforePRedictor" calling point
9) Create a new EnergyManagementSystem:Program object and configure with EMS logic statements for determining the correct window construction to apply based on the outside surface temperature
10) Write Final Conditions Message

Measure Testing

Exercise the completed measure against these test models, using the "Always Run" measure application method:

1) DOE Reference Building generated via "Create DOE Prototype Building" OpenStudio Measure. Use arguments:

    a. Building Type: Small Office
    b. Template: 90.1-2004
    c. Climate Zone = 5A (Chicago)

Required Unit Tests

None