SANDIA REPORT
SAND2016-8983
Unlimited Release
September, 2016

Institutional Transformation (IX) 2.5 Building Module Help Manual

Daniel Villa,
Sandia National Laboratories

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: http://www.ntis.gov/search

SAND2016-8983
Unlimited Release
September, 2016

Institutional Transformation 2.5 Building Module Help Manual

Daniel Villa
06926 Earth Systems Analysis

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-1138

Abstract

The Institutional Transformation (IX) building module is a software tool developed at Sandia National Laboratories to evaluate energy conservation measures (ECMs) on hundreds of DOE-2 building energy models simultaneously. In IX, ECMs can be designed through parameterizing DOE-2 building models and doing further processing via visual basic for applications subroutines. IX provides the functionality to handle multiple building models for different years, which enables incrementally changing a site of hundreds of buildings over time. It also enables evaluation of the effects of changing climate, comparisons between data and modeling results, and energy use of centralized utility buildings (CUBs). IX consists of a Microsoft Excel® user interface, Microsoft Access® database, and Microsoft Excel® CUB build utility whose functionalities are described in detail in this report. In addition to descriptions of the user interfaces, descriptions of every ECM already designed in IX is included.

Table of Contents

1 Introduction 10
1.1 History 10
1.2 Approach 10
1.3 Data Model 10
1.3.1 Energy Conservation Measures 10
1.3.2 Buildings 11
1.3.3 Weather 12
1.3.4 Warnings 12
1.4 Important Things to Know 13
2 The IX User Interface 15
2.1 Scenario Sheet 15
2.1.1 Scenario List Box 15
2.1.2 Scenario Reset Sizes Button 17
2.2 Input Sheet 17
2.2.1 Reset Sizes Button 18
2.2.2 Scenario Combo Box 18
2.2.3 ECM Operation Radio Button 18
2.2.4 ECM Combo Box 19
2.2.5 Begin Year Combo Box 19
2.2.6 End Year Combo Box 19
2.2.7 Building Selection Additional Attribute Combo Box 19
2.2.8 Select Buildings Button 20
2.2.9 Building Selection List Box 24
2.2.10 Propagate Entries in Data Table Check Box 24
2.2.11 Edit Default Year Data Check Box 25
2.2.12 Commit Changes to Database Button 25
2.2.13 Parameter Combo Box 25
2.2.14 Data Entry Table Additional Attribute Combo Box 25
2.2.15 Data Entry Table 25
2.3 Run Sheet 26
2.3.1 Reset sizes button 27
2.3.2 Scenario Combo Box 27
2.3.3 Keep Sim-files Check Box 27
2.3.4 Run in Back Ground Check Box 27

2.3.5 Scenario Information List Box ... 28
2.3.6 Review Input Button ... 28
2.3.7 Select Output Button ... 28
2.3.8 Run Button ... 29
2.3.9 Outputs Selected List Box ... 29
2.3.10 Check Back Ground Run Button ... 30
2.4 Results Sheet ... 30
2.4.1 Scenario Combo Box ... 30
2.4.2 Reset Sizes Button ... 30
2.4.3 Additional Attribute Combo Box ... 31
2.4.4 Units Combo Box ... 31
2.4.5 Saved Configurations List Box ... 31
2.4.6 Save Configuration Button ... 31
2.4.7 Delete Configuration Button ... 31
2.5 Pivot Tables ... 31
3 The IX Database ... 38
3.1 Common Functionality in Database Forms ... 39
3.2 Buildings ... 40
3.3 Weather ... 45
3.4 Energy Conservation Measures (ECMs) ... 46
4 CUB Utility ... 52
Appendix A ECM Descriptions ... 56
Air Management in IT Rooms and DCs ... 56
Airside Economizer Control ... 57
Boiler Thermal Efficiency ... 60
Caulk & Weatherstrip Doors & Windows ... 61
Chilled Water Temperature Reset ... 62
Hot Water Temperature Reset ... 64
Cool Roof ... 65
CUB Chilled Water Temperature Reset ... 66
CUB Chiller Efficiency ... 67
CUB Cooling Tower Fan Control ... 68
CUB Flat Plate Economizer Use ... 69
Daylighting Sensors for Top and Side Lighting ... 71
Duct Static Pressure Reset (repair & commission) ... 73

Exterior Insulated Finish System (EIFS) 75
High Efficiency Lighting Replacements....76
Hot Water Pump Flow Control....79
Install Exterior Shading Overhangs 81
Install Interior Shading....83
Install Revolving Doors or Vestibules....86
Insulate Roof....88
Lab Exhaust....89
Limit Personal Space Heater Use....91
New Window or Glass Properties....94
Night Cooling....98
Reduce Domestic Hot Water Recirculation Hours....102
Reduce Fan Operation Hours....103
Reduce Illumination Levels....113
Reduce Plug Loads....114
Seal Vertical Shafts & Stairways....115
Supply Air Temperature Reset....116
Thermostat Management (temperature set-back and occupancy set-point)....126
Variable Air Volume (VAV) Box Occupancy Sensors....135
Appendix B User Interface Tutorial....139
Distribution....153

List of Figures

Figure 1. Compact and repair database....14
Figure 2. Scenario sheet controls....15
Figure 3. Scenario list box right click....16
Figure 4. Input sheet controls....18
Figure 5. Hidden delete ECM button displayed in delete mode....19
Figure 6. Building selection form description....21
Figure 7. Preselected groups combo box....22
Figure 8. Group name message box....22
Figure 9. Delete group user form....23
Figure 10. Data being entered for building 6526 in 2014....24
Figure 11. No propagation....24
Figure 12. Propagate entries option causes input to propagate to end year (2020)....24
Figure 13. Prominent features of the data entry table....26

Figure 14. Data validity protection 26
Figure 15. Reset sizes button....27
Figure 16. Review input sheet 28
Figure 17. Select output form....29
Figure 18. Run progress meter 29
Figure 19. Results sheet controls....30
Figure 20. IX results pivot cache fields 32
Figure 21. Pivot table interface in Excel 33
Figure 22. Drag and drop regions 34
Figure 23. Output name as report filter 35
Figure 24. Two layers of row fields....36
Figure 25. Flipped month and year pivot table 36
Figure 26. Changing field settings....37
Figure 27. IX database table relationships....38
Figure 28. IX database main form....39
Figure 29. Database form record navigation 40
Figure 30 Building form 41
Figure 31. Building file form 42
Figure 32. Associated buildings must be checked out first....43
Figure 33. Managing building file time histories 44
Figure 34. Check in failure due to time interval overlaps....45
Figure 35. Weather file form 45
Figure 36. Weather history form 46
Figure 37. ECM form....47
Figure 38. ECM parameters form 47
Figure 39. Parameter master list form 48
Figure 40. Select from combo box....48
Figure 41. Add master list parameters to user input parameter list 49
Figure 42. Character to delimit parameter names 50
Figure 43. Inconsistent formula....50
Figure 44. CUB utility input screen 52
Figure 45. Example cooling SAT set-point as a function of outdoor air temperature 117
Figure 46. Example heating SAT set-point as a function of outdoor air temperature....118
Figure 47. Enable content for VBA....139
Figure 48. Creating a new scenario 140
Figure 49. Enter a new scenario name 140
Figure 50. Right click on an existing scenario 141
Figure 51. Edit page upper half....142
Figure 52. Selecting an ECM 142
Figure 53. Select begin and end year....143
Figure 54. Building selection form features....143
Figure 55. Buildings selected 144

Figure 56. Building information listbox 144
Figure 57. Selecting an ECM Parameter 145
Figure 58. Data Entry Features 145
Figure 59. Getting to the run sheet 146
Figure 60. Run sheet features..147
Figure 61. Input pivot table 148
Figure 62. Select output form..148
Figure 63. Progress meter 149

List of Tables

Table 1. Building file additional attribute descriptions..19
Table 2 Example rank distribution..135

1 Introduction

1.1 History

The development of the institutional transformation (IX) software package started with a vision to make it possible to make site-wide unbiased[1] analytical data part of the planner's decisions concerning investment in buildings. IX 1.0 was completed in spring 2013 and was capable of analyzing 27 office buildings for several energy conservation measures (ECMs) and IX 2.0 was delivered fall 2013 and handled 101 buildings of several types (offices, light labs, heavy labs, warehouses, etc..) and has a generalized framework in its database to build new ECMs and building types. This third version, IX 2.5, extends IX 2.0 with a brand new user interface, multiple scenario development, central utility buildings for chilled water circulation loops (which have been pre-configured in the database), variable weather into the future, and building creation and demolition.

1.2 Approach

IX is designed as an engineering analysis tool that has emphasized capability over user-friendliness. User friendliness has been addressed as much as possible but has not received nearly as much attention. The user must understand the underlying data model and must be generally aware of issues related to whole building energy analysis in IX's engine Doe2.2. In addition, the ECMs in IX must be understood and applied correctly for valid results to be produced. If an ECM is not applied correctly to each building model involved, then the results will not be correct. Also, if an ECM is conceptually invalid, results will be erroneous. It is therefore necessary to spend time calibrating building models and validating the methods used to apply ECMs for IX parameter studies to be useful.

1.3 Data Model

1.3.1 Energy Conservation Measures

IX 2.5 is designed to create scenarios of the evolution of a large complex of buildings. A scenario must have one or more ECMs. Each ECM can have a different set of buildings and a different time interval. Each ECM consists of two sets of parameters. The first set of parameters is the user input parameters. These are the numeric values that the user can alter. The second set is parameters that must exist in building input files for the ECM to work (called eQUEST parameters). Between these two sets, a function exists that can be as simple as directly passing input to the building input files to very complex operations that involve VBA functions and queries to the IX database. A simple example comes from the cool roof ECM. The set of eQUEST parameters is a single parameter called "Roof Absorbtance" and the set of user input parameters contains two parameters called "New Cool Roof SRI" and "Percent New Cool Roof." The function between the two sets is a simple linear relationship as seen below.

\`Roof Absorbtance\` = (0.95 - 0.0075*$\`New Cool Roof SRI\`)*(1 - \`Percent New Cool Roof\`) + (0.95 - 0.0075*\`New Cool Roof SRI\`)*\`Percent New Cool Roof\`

[1] This is referring to a model that is unbiased by human opinion. The models held by the IX database may continue to have biases due to shortcomings in the modeling in Doe2.2, lack of correct calibration, and inaccuracies in the method used to create the ECMs in IX. All of these factors have to be given sufficient attention to make IX a useful tool.

Here, "$" signifies that the variable after "$" will not change with time but will rather keep its original initial value for the first year of simulation.

Adding an ECM to IX involves several steps. The first is to design the ECM by specifying its user input parameters, eQUEST parameters, and functions between the two sets. The locations that an ECM should affect a building model need to be specified with instructions concerning what commands and keywords in BDL must have expressions inserted. For example, the cool roof ECM in the current IX database one can write,

For every CONSTRUCTION command used by a roof surface, insert the following

```
ABSORPTANCE     = {#pa("Roof Absorbtance")}
```

This set of instructions can become significantly more complicated depending on the intent of the ECM being designed. Once the ECM is designed, it should be applied to a single building model and tested. Ideally, the ECM should be able to be tested for several documented cases where actual changes were made to the real building with weather and energy data recorded before and after. Without such data there is no way to make an assessment of the accuracy of the ECM to real world energy savings. More often, it is necessary to carefully evaluate the ECM for reasonableness based on general statements about the effectiveness of ECMs in the literature and on expert knowledge of the accuracy of the modeling technique being used in Doe2.2 for the commands being changed. Once the ECM has been tested, the resulting parameters have to be placed in the IX database. After this, the final step is to add the relevant Doe2.2 expressions in every building input file in the IX database for which the ECM is to be added. This is a critical step, which is currently accomplished by manually altering every file. This requires that every file in the IX database be checked out. It is extremely important to perform automated checks, which count locations that an expression has been added because it is very easy to partially or incorrectly place parameter expressions. Often sophisticated search and replace operations using regular expressions can automate adding a new ECM to many files. It is also extremely important to gather the data needed to set each parameter to an accurate value based on the actual building's materials, systems, and operations.

1.3.2 Buildings

Buildings can be represented by one or more models. Each model must occupy one or more consecutive calendar years. This allows the following four events to be included in the IX data model:

1. Building demolition (i.e., a building ceases to exist after 2020)
2. Building creation (i.e., a building begins to exist in 2016)
3. Building renovation (i.e., changes that are too complicated to model using ECMs)
4. Historical calibrated models (i.e., retention of configurations that provide historical information but are no longer accurate because building configurations are constantly changing).

This flexibility allows the evolution of a site to be recorded and tracked by IX over decades. The framework for this data model is fully developed but requires operations in the IX database. Building models have to be checked in and checked out for changes to be made. A building has to have all of its individual models checked out to check it in and out of IX. Currently, the check in process performs three critical checks:

1. Search for the required parameters for each ECM and to generate the list of ECMs that the building model contains.
2. Check to be sure these parameters occur more than once in the file. One time parameters indicate that only the parameter declaration exists that suggests that the ECM is not valid in the file. Only one parameter per ECM is required to be used more than once.
3. Run the baseline model in Doe2.2 for every weather file which it is coupled with.[2] This serves to make sure that it is a valid building design language (BDL) file that runs successfully.

In the future, additional checks may be added such as a check that looks for a calibration sign off on the file. For now, IX contains a calibration status field in the building model table but does not require calibration for check in.

In IX many buildings can be represented by a single central utility building (CUB) model. CUBs are buildings that house centralized cooling and heating equipment that service many buildings. Because of CUBs, there is a many to many relationship between buildings and models in IX (i.e. 1 building can own many models and 1 model can own many buildings). Currently only chilled water loops are supported for CUBs in IX. There are significant manual operations which have to be accurately executed to create a CUB model which includes multiple buildings. A description of these steps is provided in section 4.

1.3.3 Weather

Weather histories can be created in IX that simulate changes in weather over each year. The weather file history can be viewed from the IX User interface but is controlled in the IX database. This allows users to assess the impact of climate on a site's energy use and to retain historical weather data for past years. When the weather file changes from the weather file used in the baseline year, energy savings are assessed by creating a new run of the baseline year building model with the current weather file. This keeps energy savings normalized to weather.

1.3.4 Warnings

The data model in IX 2.5 has many more features than 2.0 and is therefore much more complicated. Like any other complicated engineering tool, it is therefore easy to create input in IX 2.5 that the user does not understand. This can lead to a naïve user reporting results whose inputs do not actually match their own internal idea about what they input. With hundreds of building files (models) and hundreds of input parameters per file, no human mind can keep up with all of the input for a complicated scenario. Documentation of the intent of a scenario should therefore be maintained to enable auditing of actual input versus stated intent. Results should also therefore be retained for a sufficient time to be sure that a scenario can be reviewed for accuracy to troubleshoot if actual implementation is considerably different than the estimates provided by IX. The following situations are common pitfalls when making input into IX:

[2] Variable weather can be implemented. The weather input files have to be checked for validity also and there is the rare chance that coupling a building file and a weather file will cause a bug in Doe2.2 which was not captured by individual checks.

1. Users select different year intervals for two ECMs and expect input to propagate. For example suppose a user chooses 2013-2014 for ECM1 and enters data. The user then chooses 2015-2016 for ECM2 and enters data. At this point the user may think that a continuous record of change has been made but is surprised to find that energy savings do not increase very much in 2015. The problem is that the buildings selected for ECM1 have returned to their baseline values in 2015 which results in 0 energy savings due to ECM1 even though the energy savings were present for 2014. It is therefore often best to select the same time interval for all ECMs.

2. Users are confused by results because they do not recognize that a building has transitioned from one model to another model and do not know the differences between the two models.

3. Users do not enter information in every parameter needed for an ECM, which leaves the intended input incomplete. For example, a user may enter a cool roof solar reflective index but leave the percentage application to the new roof equal to 0%. This can be amended by reviewing the actual building model inputs (eQUEST parameters), which are available as "ready only" tables.

1.4 Important Things to Know

There are several facts that are important to know when using IX.

1. IX is a stand-alone application. This means that the database is not centralized such that updates will be made only to the user's copy of the database.

2. The latest revision of the user interface and database are contained in a Team Forge repository. Access to the project can be obtained by contacting Daniel Villa (dlvilla@sandia.gov)

3. Many revisions of IX are stored in a share drive at \\snl\Collaborative\IX\version2_5.

4. Any significant results from IX used to make decisions should be stored in \\snl\Collaborative\IX\Results. The entire set of files for the version of IX used should be copied to this location.

5. There are special files called Details files that are maintained in .\IXData\BuildingDetailedInformation and contain important information about assumptions and calibration of building files. A details file may or may not exist for a given building input file.

6. Most of the time IX automatically accomplishes a compact and repair but sometimes the routine to do this fails. If this starts occurring, the IX database will grow in size fairly quickly as tables are altered by the IX user interface. It has to be compacted and repaired fairly often to keep its size down. To do this, go to ./IXData/ and open IXDatabase2_5.accdb. Navigate to the file tab and press "Compact & Repair Database" as seen in Figure 1.

Figure 1. Compact and repair database

7. When configuring CUBs using the CUB Utility it is necessary to manually update the tblCUBNameChanges in the database. After using the CUB Utility, copy the entries in the "Parameter Name Table" sheet to the "ListOfAllParameterNamesChanged" table. If there are already entries for the Building ID being worked with, then it may be necessary to delete some of the entries in "Parameter Name Table." Delete any entries that are older and overlap with the building ID and begin and end year of each row so that there are no duplicate entries. Once this has been accomplished. Delete all of the entries in "tblCUBNameChanges" and paste in the list in "ListOfAllParameterNamesChanged." If this is not done every time CUBs are changed, the CUBs may not have ECMs added that are present.

2 The IX User Interface

The IX user interface is a set of six Excel spreadsheets with various controls used to create a scenario. Its primary purpose is to create scenarios of a "static configuration" of an entire site. Here, "static configuration" refers to the fact that it is assumed that the database is not changed during the development of scenarios. IX 2.5 does not have the necessary programming to enforce consistency between scenarios and database configuration. It is therefore necessary that new scenarios be developed whenever significant changes are made to the database since there is no guarantee of consistency. Changing the site configuration occurs in the IX database interface, which is covered in the next chapter.

2.1 Scenario Sheet

When opened, the IX user interface always starts at the scenario sheet. This sheet provides a list of all of the scenarios that have been created by the user along with basic information about the scenario. Two controls exist on this sheet.

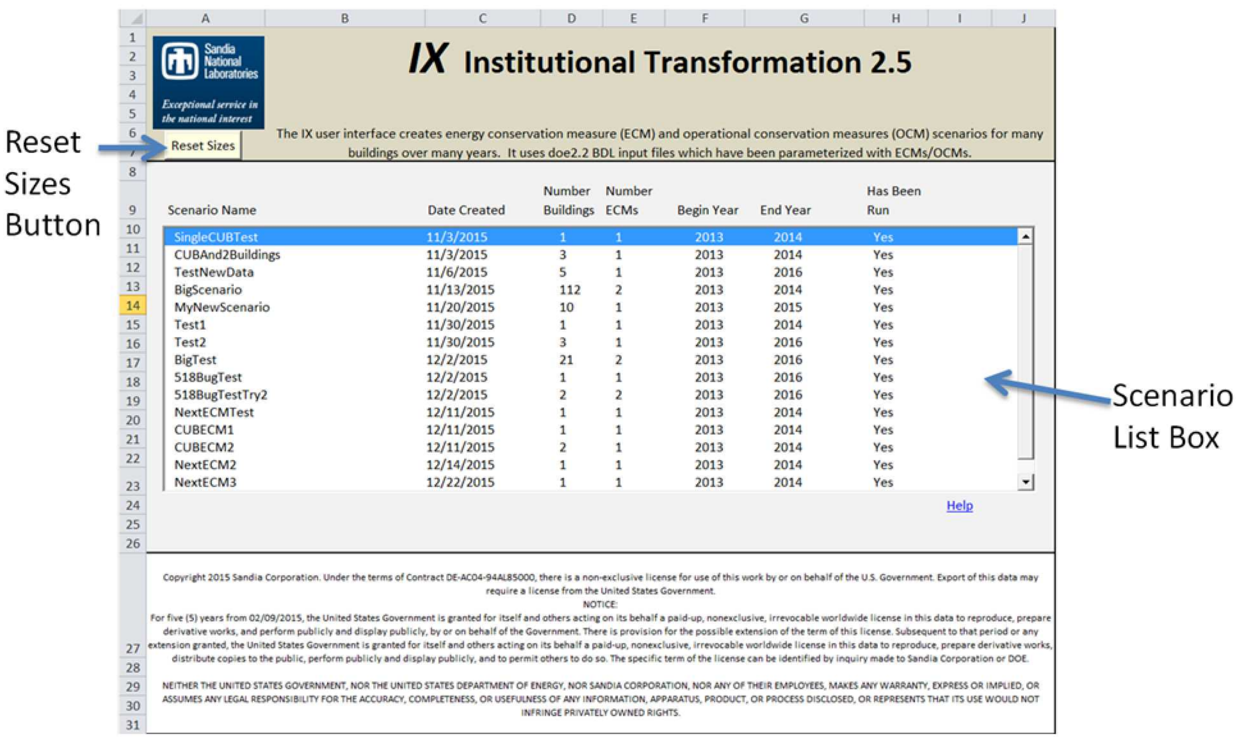


Figure 2. Scenario sheet controls

2.1.1 Scenario List Box

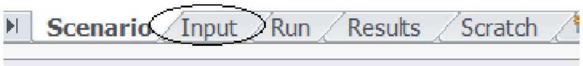
A right-click on the list box, as seen in Figure 3, produces a menu with six options.[3,4]

- **New** – Create a new empty scenario. Enter a name with no spaces. Only Alphanumeric characters and "_" are allowed.

[3] Only the "New" option is presented if no scenario exists or if the user selects a row in the list box that does not contain a scenario.

[4] Sometimes only the menu outline appears. The user has to move the mouse over the menu to see the options. This is an unfortunate bug beyond the scope of the IX program.

- **Edit** – Navigate to the “Input” sheet. The current selected scenario will be selected. This is equivalent to pressing the “Input” tab traditionally used to switch sheets in Excel.



- **Run** – Navigate to the “Run” sheet. The current scenario will be selected. This is equivalent to pressing the “Run” tab traditionally used to switch sheets in Excel.[5]
- **Results** – Navigate to the “Results” sheet. The current scenario will be selected. This is equivalent to pressing the “Results” tab traditionally used to switch sheets in Excel.[6]
- **Delete** – Delete the selected scenario. Double confirmation is required afterward to avoid a mistake. This is a permanent deletion.
- Copy – Copy the selected scenario with a new name. This will replicate all input and results but will not create a new copy of the files in ./IXData/Results/<Scenario Name>. This function is very useful when you want to make a change to a scenario while keeping a copy of the original configuration.

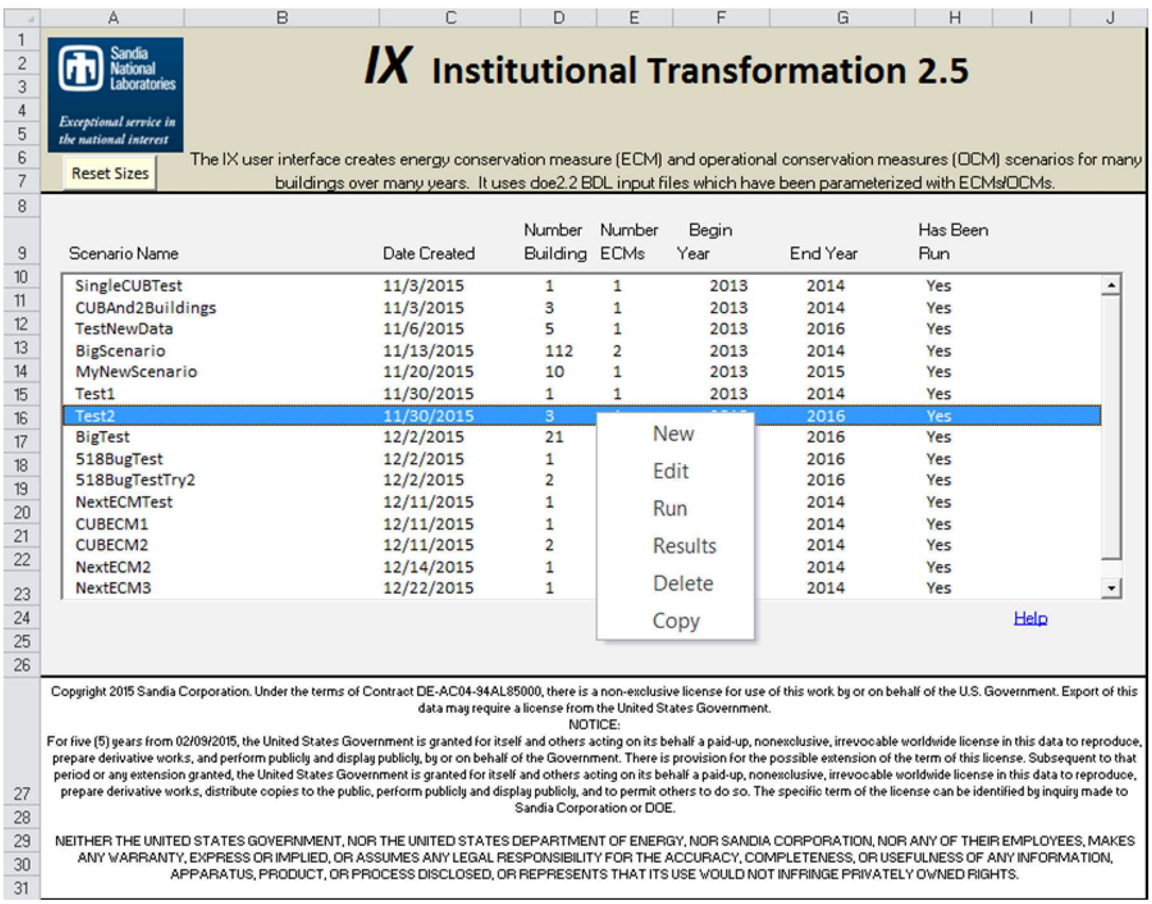


Figure 3. Scenario list box right click

[5] If the scenario is empty, then a warning will appear stating that the scenario is either empty or corrupt. For this case navigate to the “Input” sheet to add data to the scenario.
[6] If the scenario has not been run, then a warning will appear and no action will take place.

2.1.2 Scenario Reset Sizes Button

The only other options on this sheet are to select help, which opens this file, and to press the "Reset Sizes" button. Press this button if any of the controls seem disproportionately large or small. The need for this button is due to a bug in Excel VBA's architecture that occurs when the graphics for a monitor are in a non-native setting. This often occurs for Web-meetings or projectors. Under these conditions the controls in IX may behave erratically and shift or change size in a way that renders the application inoperable (i.e. a control will often grow to the point of covering the entire sheet so that nothing else can be selected). The only fix that worked every time was to place a table in the database of all of the control sizes and positions and to create a function that reset them through a manual click.

2.2 Input Sheet

The input sheet is the location where most work is performed in the IX user interface. It provides the means to enter and change data input into a scenario. If a scenario has already been run, this sheet can be used for reviewing input. Changes can still be made but changing the begin or end year, adding an ECM, changing the selection of buildings, or changing any of the input data cannot be accomplished unless the results for that scenario are deleted (warning message boxes will prompt the user[7]). This enforces consistency by preventing a scenario that has input data different from the input data that existed when the scenario was run. To avoid overwriting results, the user can copy a scenario and then make changes.

There are 16 controls on the input sheet which makes it the most complex sheet in IX as seen in Figure 4. Refer to this figure to locate controls being referred in the coming sections. The controls are arranged in such a way that the user is expected to start at the top and work down. Any of the controls can be changed at any time but such changes can result in undoing previous work (i.e., if you change the time interval all data input for the years that are no longer present will be deleted). Regardless, the user will be warned about consequences of specific actions and will have the chance to undo them.

[7] There is more than one layer of warning messages. Each layer cannot be undone once the user has made a decision. This can lead to undesirable results because a user may want to undo a previous decision in light of the next layer's warning but keeping the messages to one layer would require a lot of rework.

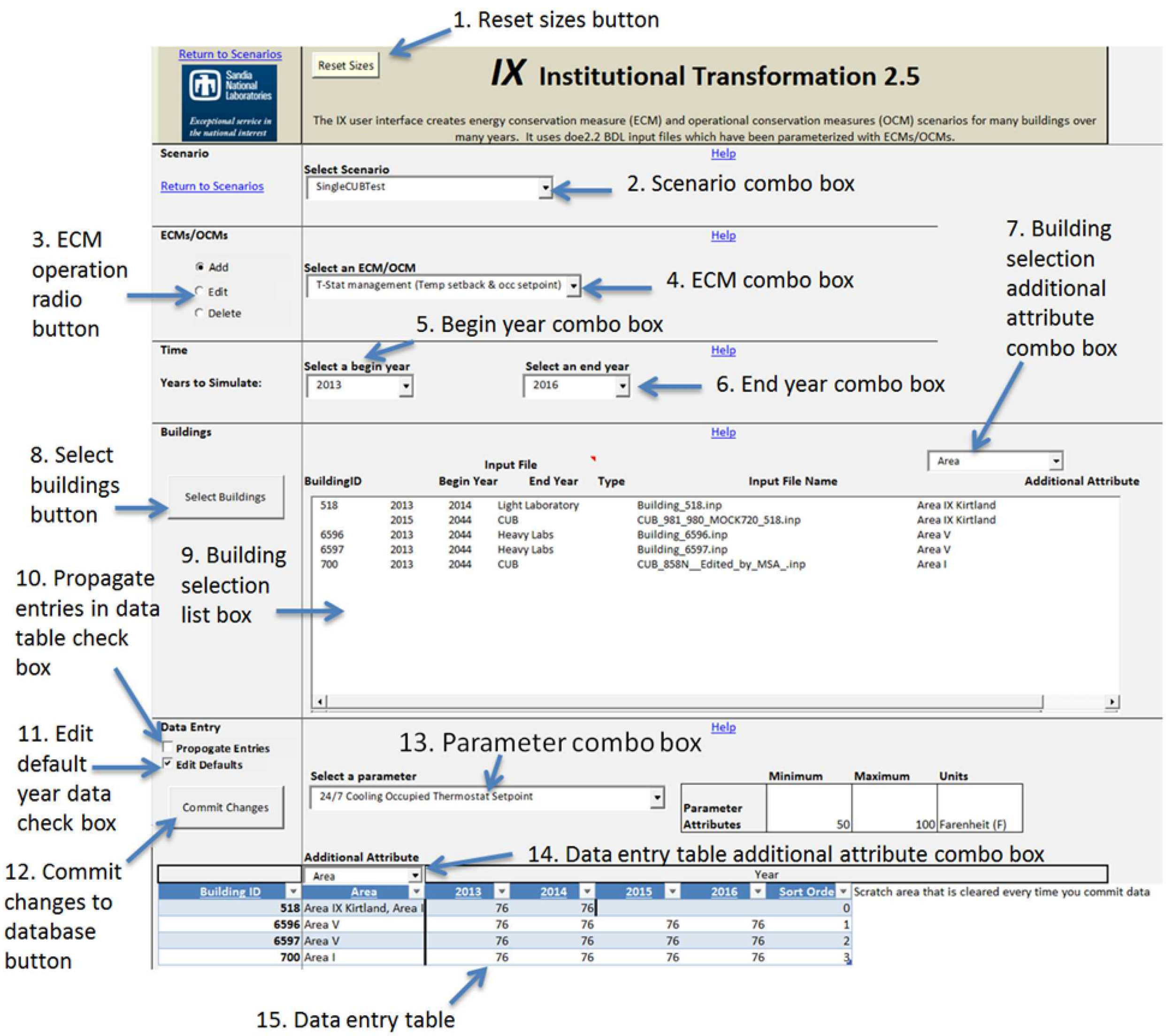


Figure 4. Input sheet controls

2.2.1 Reset Sizes Button

Press this button whenever any of the controls seem misplaced or have grown larger or smaller. This control is due to a Microsoft bug. See section 2.1.2 for a complete explanation.

2.2.2 Scenario Combo Box

This combo box provides a convenient way to switch between scenarios without having to return to the scenario sheet.

2.2.3 ECM Operation Radio Button

This radio button switches between three modes of operating on energy conservation measures (ECM). When a scenario is empty, the only operation possible is to “Add” an ECM. The entire list of ECM’s in the IX database is available in the ECM combo box for an empty ECM. Once data has been committed, the mode automatically switches to “Edit.” After this, if a new ECM is desired, selecting “Add” again will repopulate the ECM combo box with every ECM except the ECM that has already been added. In general, there will always be a list of ECMs that have already been added and can be accessed in the "Edit" mode while a second list of ECMs that have not been added can be accessed through the "Add" mode. If an ECM already added needs to be deleted, selecting "Delete" mode will cause a delete ECM button to appear next to the ECM combo box (4) as seen in Figure 5.

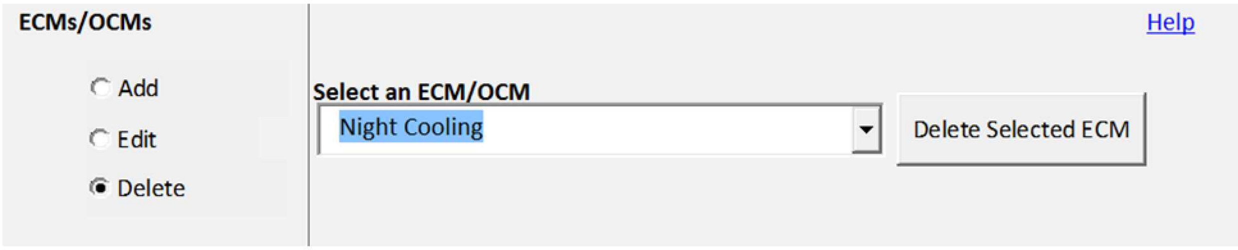


Figure 5. Hidden delete ECM button displayed in delete mode

Sometimes the mode will not switch from "Add" to "Edit" on the first click. This bug has not yet been able to be resolved and is harmless. A second click will successfully change the mode.

2.2.4 ECM Combo Box

This combo box allows selection of the ECM to be Added, Edited, or Deleted. The list of ECMs is dependent on whether an ECM has already been added through committing parameter changes.

2.2.5 Begin Year Combo Box

This combo box allows the user to set the begin year for a specific ECM. The begin year can be different for different ECMs but it is recommended that the same begin and end years be used across all ECMs for a scenario.

2.2.6 End Year Combo Box

This combo box allows the user to set the end year for a specific ECM. The end year can be different for different ECMs but it is recommended that the same begin and end years be used across all ECM's for a scenario.

2.2.7 Building Selection Additional Attribute Combo Box

This combo box controls display of an additional attribute in the right-most column of the building selection list box. This allows the display of additional information in the database. The additional attributes available are described below

Table 1. Building files additional attribute descriptions

Building Attribute	Description
Area	A group of buildings in a common area. (Tech area I, II, etc.)
Building File Changes	Notes on how a building file has changed from previous files
Calibration	Statement concerning calibration status of a building file.

Building Attribute	Description
Details File	A file reference (if any) in ./IXData/BuildingDetailedInformation that contains detailed information about the building's calibration, lighting, and underlying assumptions in the building file
Site	A larger grouping than Area that designates an entire geographic location. For example, SNL has the "New Mexico" and "California" sites
Square Feet (ft2)	Number of square feet in the building file
Type	Building type (Auditorium, Cafeteria, CUB, Data Center, Education, Heavy Labs, Light Laboratory, Medical, Office, Semi-Conductor Fabrication, Warehouses). This designation is only significant when the building type is CUB otherwise it is only a way of classifying a building file.
Volume (ft3)	Number of cubic feet in the building file
Weather History	Weather history for the building (weather history is assigned at the building level rather than the building file level)

2.2.8 Select Buildings Button

Clicking this button opens a form enabling the user to select a set of buildings as seen in Figure 6. Refer to Figure 6 in the coming sections to locate the correct control.

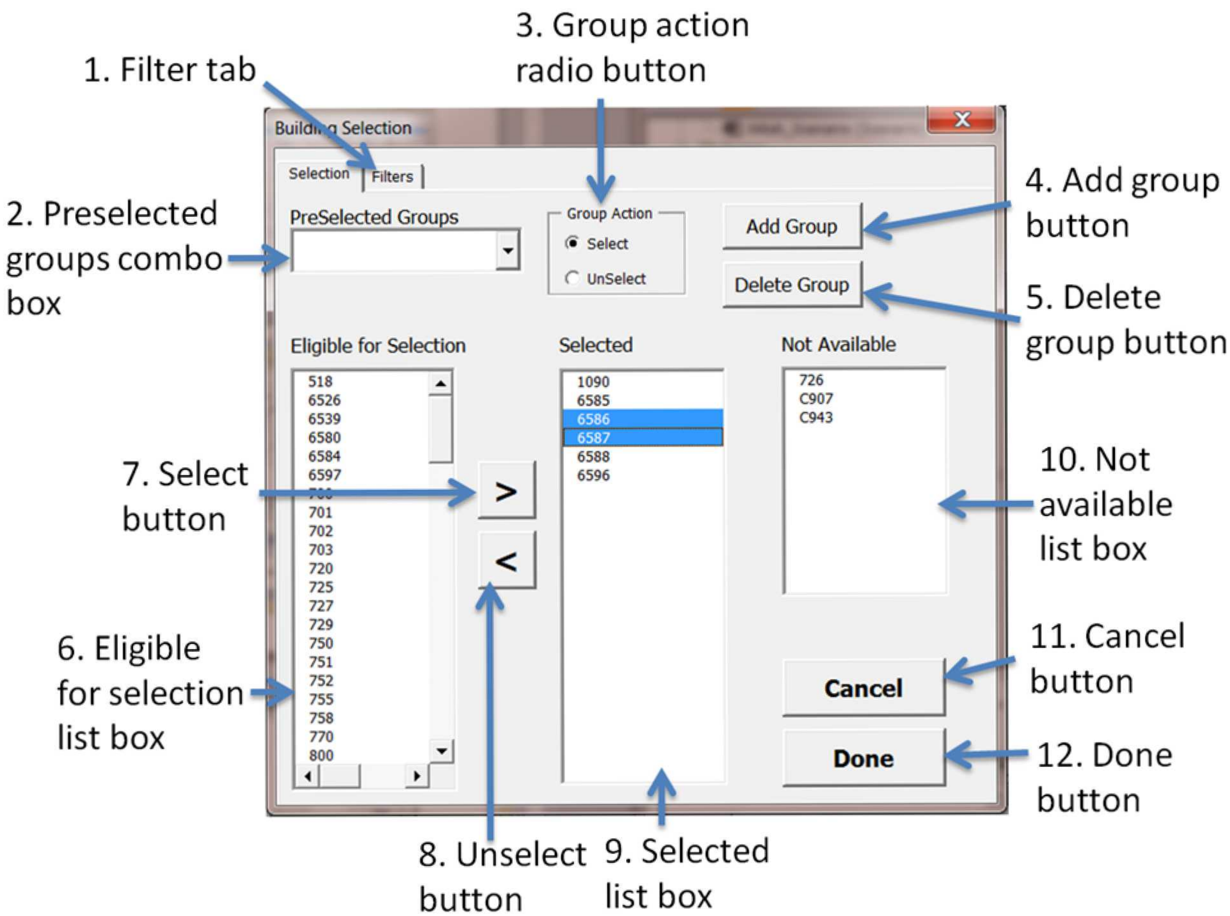


Figure 6. Building selection form description

2.2.8.1 Filters Tab

This feature is non-functional in IX 2.5

2.2.8.2 Preselected Groups Combo Box

This combo box shown in Figure 7 provides a list of groups that have been created using the "Add Group" button. Each group is a set of buildings that can either be added or taken away from the current set in the selected list box. This gives a way for the user to create sets of buildings that can be quickly selected repeatedly. Selecting an entry in the combo box list will immediately select or unselect the specified set of buildings depending on the state of the group action radio button.

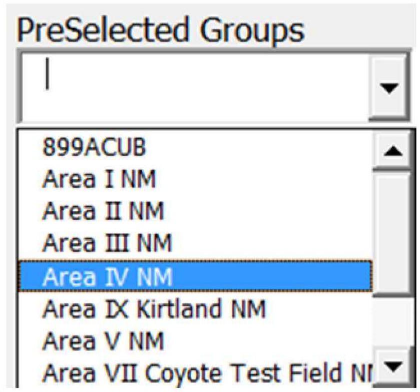


Figure 7. Preselected groups combo box

2.2.8.3 Group Action Radio Button

This radio button toggles between selecting buildings and unselecting buildings whenever the preselected group's combo box is used.

2.2.8.4 Add Group Button

This button adds the current set of buildings in the selected list box to a new group. The user is prompted to enter a name for the group as seen in Figure 8.

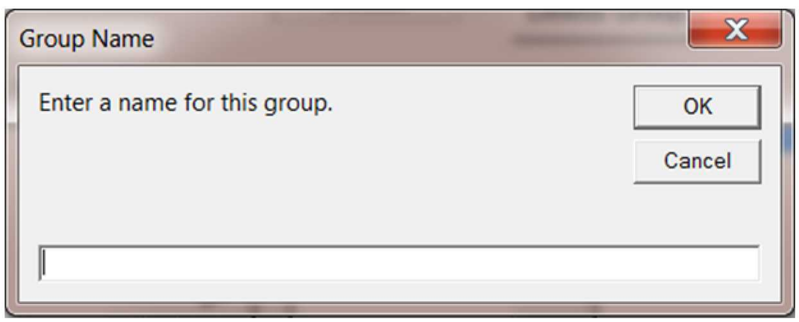


Figure 8. Group name message box

2.2.8.5 Delete Group Button

This button opens a form as seen in Figure 9 that allows the selection of building groups to be deleted. Select the groups that are to be deleted and press ok or press cancel to escape without deleting any groups

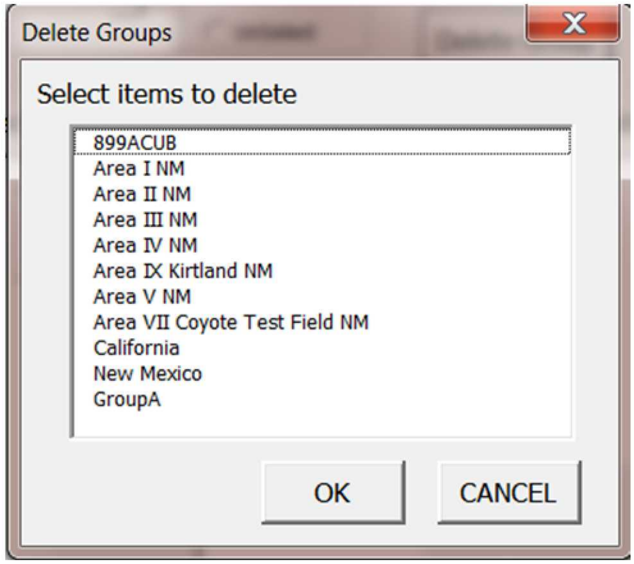


Figure 9. Delete group user form

2.2.8.6 Eligible for Selection List Box

This list box contains the entire list of building IDs eligible for selection for the current ECM and time interval. Any buildings that are not eligible are displayed in the not available list box. A selection in this list box can be moved to the selected list box by pressing the select button.

2.2.8.7 Select Button

When pressed, this button moves a selection in the eligible for selection list box into the selected list box.

2.2.8.8 Unselect Button

When pressed, this button moves a selection in the selected list box back to the eligible for selection list box.

2.2.8.9 Selected List Box

This list box contains the set of buildings that are selected for the current ECM and time interval.

2.2.8.10 Cancel Button

This button cancels all changes to building selection and exits the form.

2.2.8.11 Done Button

This button finalizes changes to building selection. After pressing this button, two issues may come up that will require further attention, if applicable. The first issue is that if there are results for the current scenario and a change has been made to the building selection, then the results will have to be deleted to keep the scenario input configuration consistent. The user will have to decide whether to cancel changes to the building selection or to delete the scenario results. The second is that if buildings have been deselected, then input data for those buildings will be erased. The user must decide whether to delete the input or to cancel changes to the building selection.

2.2.8.12 Not Available List Box

This list box gives all buildings that cannot be selected because they either do not have the ECM that is currently selected or because they do not have the ECM for the time interval selected. A building is selectable if it has the ECM for at least one year in the time interval selected.

2.2.9 Building Selection List Box

Once a set of buildings has been selected, this list box provides a summary of information concerning the selected buildings. Each building file that is within the time interval selected is listed along with begin year, end year, building type, and input file name. An additional attribute can be listed using the building selection additional attribute combo box as described in section 2.2.7.

2.2.10 Propagate Entries in Data Table Check Box

When selected, this check box causes data to propagate to the end year. This can reduce redundant data input. Figure 10, Figure 11, and Figure 12 illustrate the difference between data entry without and with propagation.

	Area	Year							
Building ID	**Area**	**2013**	**2014**	**2015**	**2016**	**2017**	**2018**	**2019**	**2020**
1090	Area II	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
518	Area IX Kirtland; Area IX	1	1	1	1	1	1	1	1
6526	Area III	1	0.5	1	1	1	1	1	1
6539	Area III	2	2	2	2	2	2	2	2
6580	Area V	1	1	1	1	1	1	1	1
6584	Area III	1	1	1	1	1	1	1	1
6585	Area V	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
6586	Area V	1	1	1	1	1	1	1	1
6587	Area III	1	1	1	1	1	1	1	1
6588	Area V	1	1	1	1	1	1	1	1
6596	Area V	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

Figure 10. Data being entered for building 6526 in 2014

Building ID	Area	2013	2014	2015	2016	2017	2018	2019	2020
1090	Area II	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
518	Area IX Kirtland; Area IX	1	1	1	1	1	1	1	1
6526	Area III	1	0.5	1	1	1	1	1	1
6539	Area III	2	2	2	2	2	2	2	2
6580	Area V	1	1	1	1	1	1	1	1
6584	Area III	1	1	1	1	1	1	1	1
6585	Area V	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
6586	Area V	1	1	1	1	1	1	1	1
6587	Area III	1	1	1	1	1	1	1	1
6588	Area V	1	1	1	1	1	1	1	1
6596	Area V	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

Figure 11. No propagation

Building ID	Area	2013	2014	2015	2016	2017	2018	2019	2020
1090	Area II	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
518	Area IX Kirtland; Area IX	1	1	1	1	1	1	1	1
6526	Area III	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5
6539	Area III	2	2	2	2	2	2	2	2
6580	Area V	1	1	1	1	1	1	1	1
6584	Area III	1	1	1	1	1	1	1	1
6585	Area V	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
6586	Area V	1	1	1	1	1	1	1	1
6587	Area III	1	1	1	1	1	1	1	1
6588	Area V	1	1	1	1	1	1	1	1
6596	Area V	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

Figure 12. Propagate entries option causes input to propagate to end year (2020)

2.2.11 Edit Default Year Data Check Box

When selected, this check box allows the user to edit the first year of data for a building. Usually, the first year is locked because the building model should already contain a baseline configuration that represents the building accurately. In cases where this is not true, this option provides a quick way to update a building temporarily. The database baseline values are not changed though. Each building input file has to be checked out and changed to correctly address inaccuracies in baseline configurations of buildings.

2.2.12 Commit Changes to Database Button

Once changes have been made to the data entry table, this button commits them to the database. Commitment to the database is also accomplished by several user prompts that occur if the user tries to leave the input sheet or change the parameter being worked on before committing changes.

2.2.13 Parameter Combo Box

This combo box contains the list of user input parameters and eQUEST parameters (read only), which are part of the current ECM. For user input parameters, the user must carefully input data and commit changes to the database. Refer to the ECM descriptions in Appendix A for descriptions of each parameter. After all user input parameters have been addressed, the user can view the eQUEST parameters (read only) to verify that changes have been made to the actual input in the building input files. This is an important step because it is possible to overlook how an ECM works and input data that do not affect the eQUEST parameters as expected. A simple example is that a user might change the roof insulation user input parameter for the Insulate roof ECM but forget to change the percentage of roof to apply the new insulation value, whose default is zero. The end result is zero change. Viewing the eQUEST parameters will reveal this oversite. A more efficient method to view the input is through a pivot table accessed in the run sheet (see section 2.3).

To the right of the parameter combo box is a table listing the minimum value, maximum value, and units of the current parameter.

2.2.14 Data Entry Table Additional Attribute Combo Box

This combo box controls which additional attribute is displayed alongside the building ID's. If more than one building file represents a given building ID, a semi-colon delimited list of attributes is output. See Table 1 for a list of available attributes.

2.2.15 Data Entry Table

The data entry table is used to input data for each ECM parameter. It has the same number of rows as the number of buildings that have been selected and the same number of columns as years that have been selected. The parameter displayed is controlled by the parameter combo box described in section 2.2.13. Figure 13 provides descriptions of some of the prominent features of the data entry table. Data entry must fall between the minimum and maximum values located above the data entry table. All other input is rejected as seen in Figure 14. This keeps data input as a valid type and within a designated range. In IX parameters can only be numeric values. Once changes have been made the user can press the commit changes button to send changes to the database. Sometimes it may be easier to paste values from another Excel spreadsheet into this table.

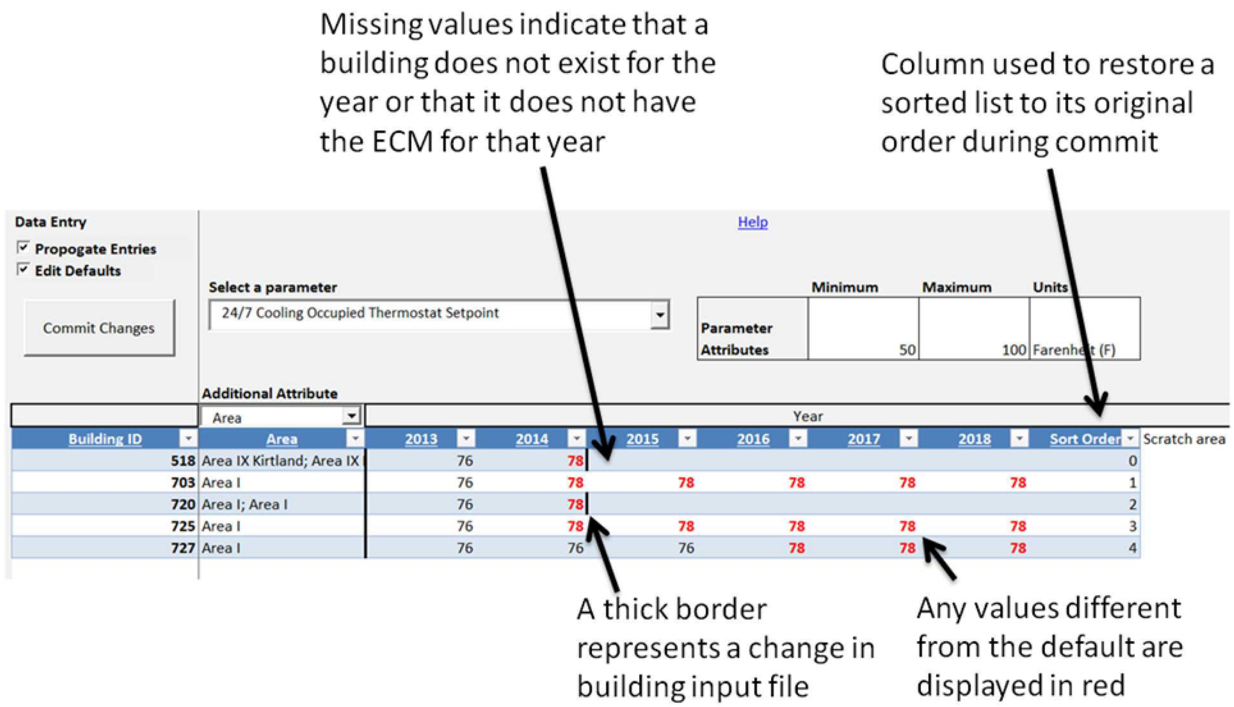


Figure 13. Prominent features of the data entry table

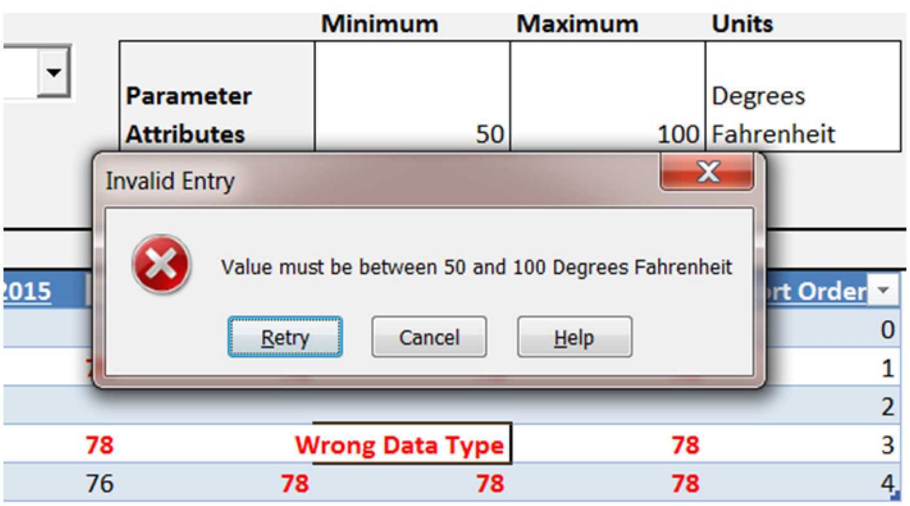


Figure 14. Data validity protection

2.3 Run Sheet

The run sheet is used to review input in a pivot table, select output, and run a scenario. Figure 15 provides a display of all of the controls referenced in the following sections.

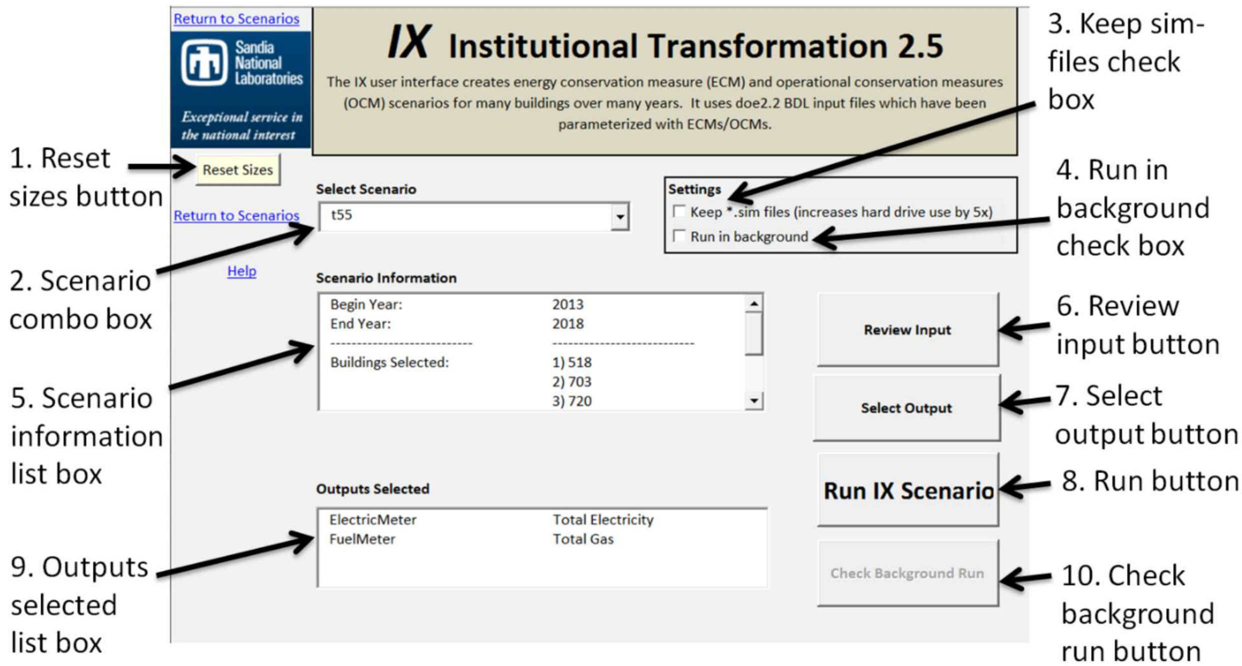


Figure 15. Reset sizes button

2.3.1 Reset sizes button

Press this button whenever any of the controls seem misplaced or have grown larger or smaller. This control is due to a Microsoft bug. See section 2.1.2 for a complete explanation.

2.3.2 Scenario Combo Box

This combo box can be used to navigate to any scenario in the database. Any scenario that is empty will cause the run sheet functions to become frozen. In this case, the user will receive a prompt indicating that the scenario is empty or corrupt.

2.3.3 Keep Sim-files Check Box

This check box provides the option to keep the simulation files for Doe2.2 runs. When not checked IX only retains hourly output files. The drawback to keeping all the files is in extra hard drive space. Keeping this box checked roughly uses five times more memory than when it is unchecked. This option is usually not needed. The files can be found in ./IXData/Results/<Scenario Name>.

2.3.4 Run in Back Ground Check Box

This option causes the Doe2.2 runs to be run in the background. IX still has to spend a considerable amount of time making changes to all of the files involved but after everything has been written, IX will use as many processors as are available minus one (to keep from overloading the system) to run Doe2.2 in several parallel processes. Once these processes start, the user can continue to use IX to develop scenarios and review results.

2.3.5 Scenario Information List Box

This list box provides a list of the scenario begin and end year, buildings selected, and ECMs selected. It exists for informational purposes only. If anything seems incorrect, then return to the input sheet to correct the input.

2.3.6 Review Input Button

Pressing this button navigates to a new sheet that displays the scenario input through a pivot table and pivot chart (see section 2.5). The user can return to this sheet at any time to verify whether a scenario's input matches expectations. This provides a powerful way to sift through complicated input, which can often be done much more quickly than through looking at the data entry tables in the input sheet (see section 2.2.15). Figure 16 provides a view of the review input sheet where several pivot table filters have been applied to reduce the data displayed to a single parameter.

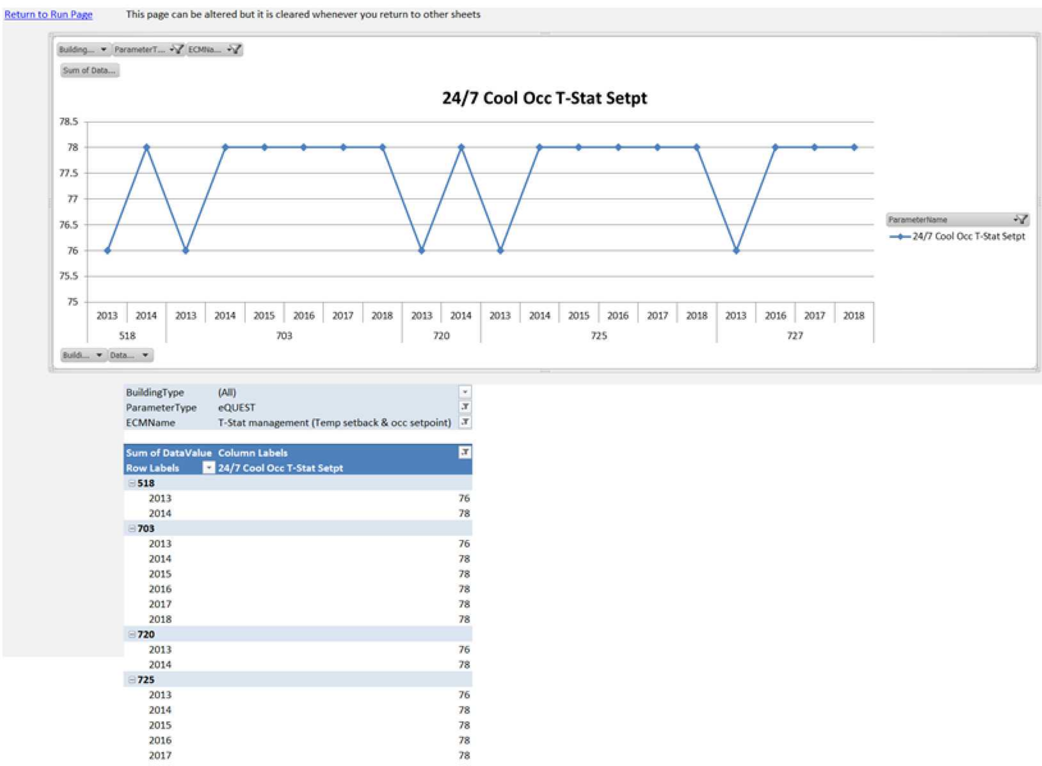


Figure 16. Review input sheet

2.3.7 Select Output Button

Pressing this button opens another form, as seen in Figure 17, which is used to select which outputs to save in the scenario. The default is to only save total electricity and total gas but the energy can be broken down further if desired. The selections are displayed in the run sheet in the outputs selected list box.

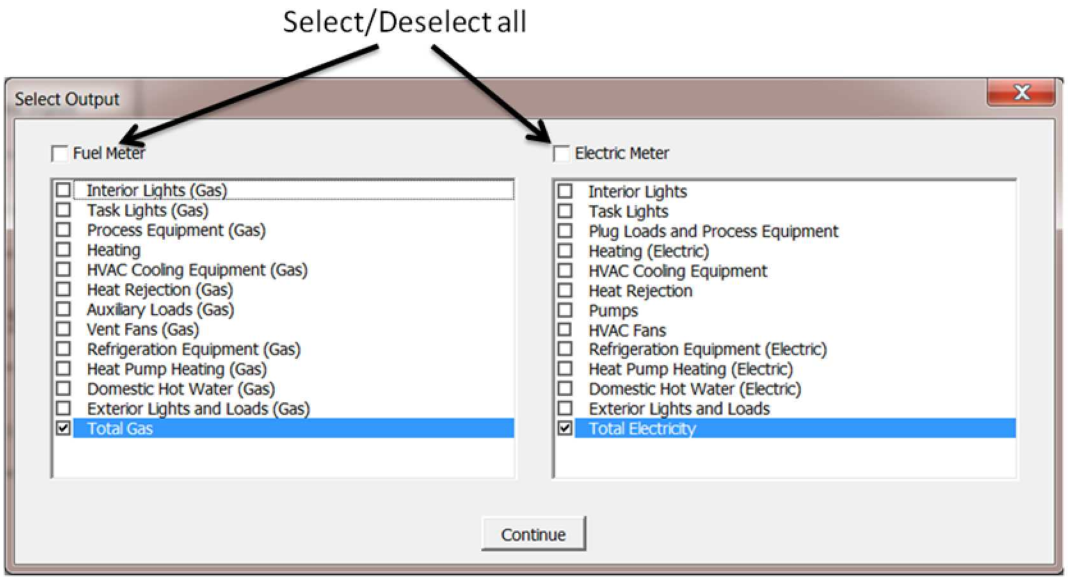


Figure 17. Select output form

2.3.8 Run Button

Pressing this button runs the IX scenario. A progress meter will pop up as seen in Figure 18 and will specify what operations are being performed. The option to cancel the scenario is available but there is often a considerable delay (up to several minutes) between pressing cancel and the actual cancelation. Also, the progress meter sometimes disappears and can only be found by minimizing and maximizing the Excel spreadsheet or minimizing all programs. The program automatically saves everything before initiating the run. If the run is in the background, the progress meter is only displayed for the file writing operations.

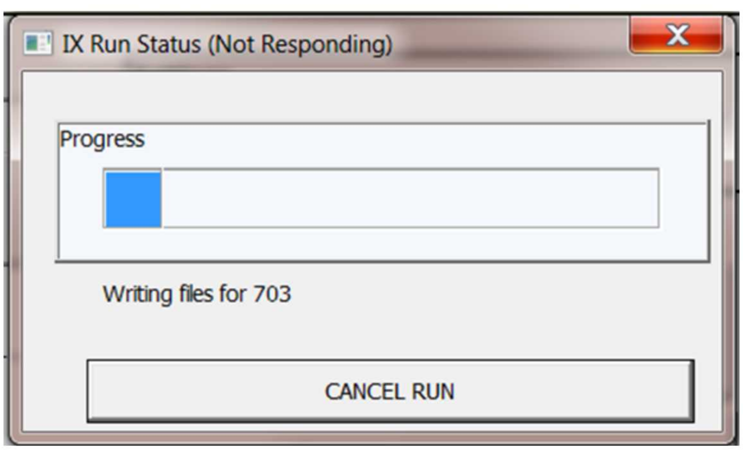


Figure 18. Run progress meter

2.3.9 Outputs Selected List Box

This list box displays which outputs have been selected. To change the outputs, press the select output button described in section 2.3.7.

2.3.10 Check Back Ground Run Button

This button initiates a function that checks to see how far along the background run has come. If the runs are complete, this button initiates post-processing of the scenario results.

2.4 Results Sheet

The results sheet is used for viewing scenario results as seen in Figure 19. Its main feature is a pivot chart and a pivot table which can be used to filter large amounts of data into a desirable form. In addition to this, there is the capability to display an additional attribute as a field, change the units being displayed, and to save configurations of the pivot table. Whenever the scenario combo box, additional attribute combo box or units combo box are changed, the entire sheet has to refresh and the pivot table and chart are returned to their default selections. The user should keep this in mind and save a configuration before changing these combo boxes so that it is easy to return to the desired configuration.

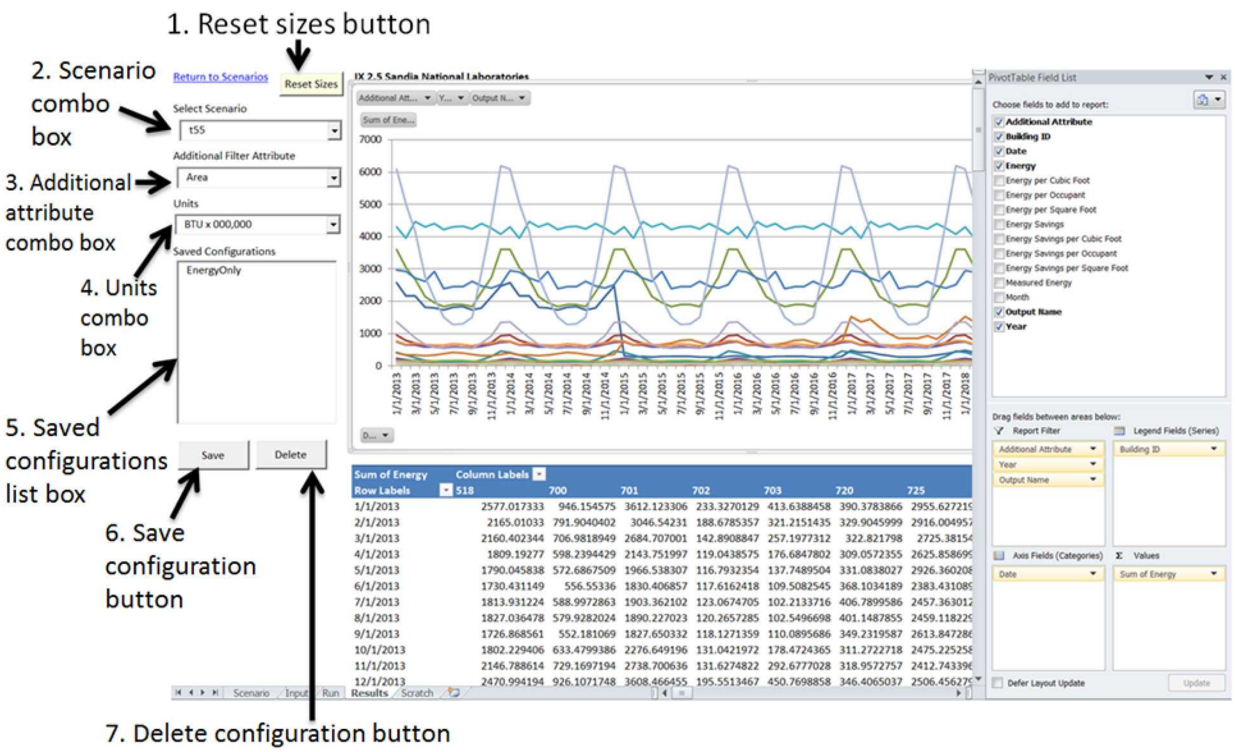


Figure 19. Results sheet controls

2.4.1 Scenario Combo Box

This enables changing to a different scenario. Only scenarios that have results are included in the list. When the scenario is changed the pivot table and chart are refreshed, overwriting the current state of the pivot table. The user should therefore be sure to save the configuration before changing the scenario.

2.4.2 Reset Sizes Button

Press this button whenever any of the controls seem misplaced or have grown larger or smaller. This control is due to a Microsoft bug. See section 2.1.2 for a complete explanation.

2.4.3 Additional Attribute Combo Box

This combo box changes the additional attribute field included in the pivot table. Changing the additional attribute requires rewriting the entire pivot table, which causes the current configuration to be lost. The additional attribute can be any of the items shown in Table 1. One of the most important applications of this option is to enable sorting of results by calibration status. It also enables selection of results by site and area.

2.4.4 Units Combo Box

This combo box enables changing the units displayed. Changing the units requires a refresh of the pivot table, causing the current configuration to be lost. The user should therefore save a configuration before changing the units.

2.4.5 Saved Configurations List Box

This list box shows all of the configurations that have been saved for the current scenario. Configurations save the chart type, field positions of the pivot table, and hidden/visible status of all fields. Any sorting or more sophisticated filtering is not included. Selecting a configuration returns the pivot table and chart to the underlying configuration.

2.4.6 Save Configuration Button

Pressing this button saves the current configuration of the pivot chart. This can take a while for larger scenarios. A progress meter is therefore used to indicate completion of the save process.

2.4.7 Delete Configuration Button

Pressing this button deletes the selected configuration in the saved configurations list box.

2.5 Pivot Tables

Pivot tables are used to display input and results in IX. A pivot table is based on a pivot cache, which contains data in a “flat” format where each row represents a complete entry with several value fields and attribute fields as seen in Figure 20.

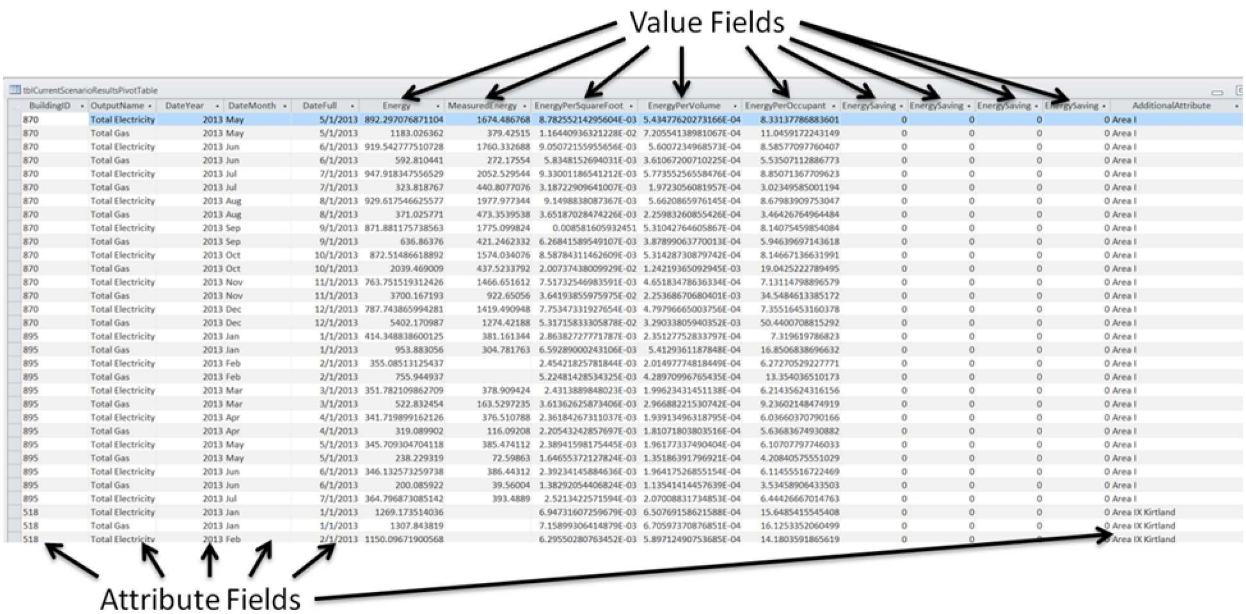


Figure 20. IX results pivot cache fields

A pivot table is a special arrangement of the fields in the pivot cache. This arrangement is controlled through the pivot table user interface in Excel as seen in Figure 21. To get to this interface, click on the pivot table or pivot chart[8] in the results sheet. If the interface does not appear, right click on the same location and select "Show Field List." This interface allows drag and drop of fields between any of the five boxes indicated in Figure 22. Even though any field can be placed in any of the boxes, it is usually most meaningful to assign numeric values as value fields and descriptive values as attribute fields.

Each field can be ignored or added to the pivot table by checking or unchecking its checkbox. Ignored fields are not included in the pivot table but can still have an effect. For example, if the year and date fields are ignored (see Figure 20), then data points with the same month are indistinguishable and therefore have a function applied to them, which is usually summation. The user must therefore exercise caution because summation of any of the normalized energy fields such as energy per square foot is an invalid operation. For these cases, results in the pivot table will be invalid unless a set of attributes fields (discussed in the next paragraph) is selected that have a one-to-one mapping between data points in the pivot cache and values displayed in the pivot table. For IX's results pivot table, a one-to-one mapping exists as long as the building ID, output name, month, and year are all specified. For energy and energy savings this is not an issue.

Added fields can be classified as attribute fields or as value fields. Attribute fields serve to categorize data. A set of attributes fully specify the meaning of a value in the pivot table. For example, for IX scenario results, a data point can be specified by building ID, output name, year, and month. If only output name, year, and month are specified as attributes, then a value in the pivot table indicates the summation (or another function) over all building IDs.

[8] For the pivot chart this discussion is the same except that column fields are called legend fields and row fields are called axis fields.

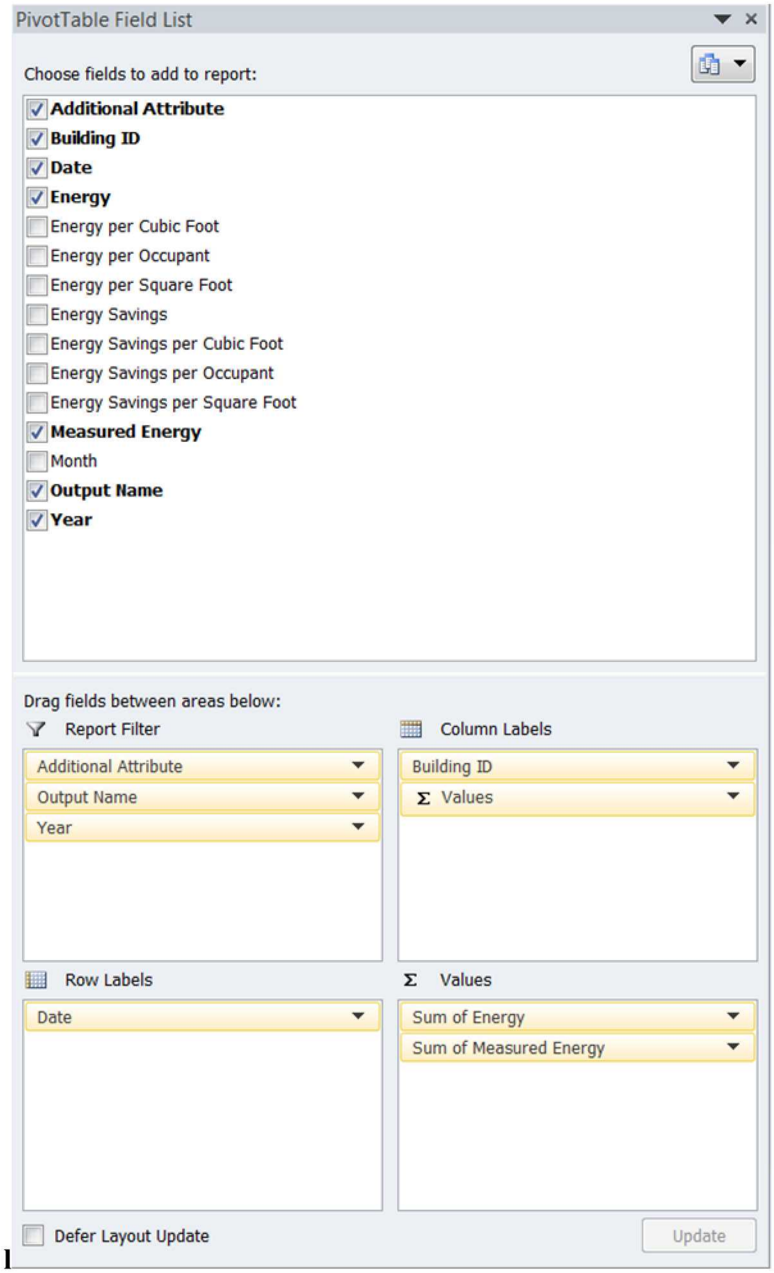


Figure 21. Pivot table interface in Excel

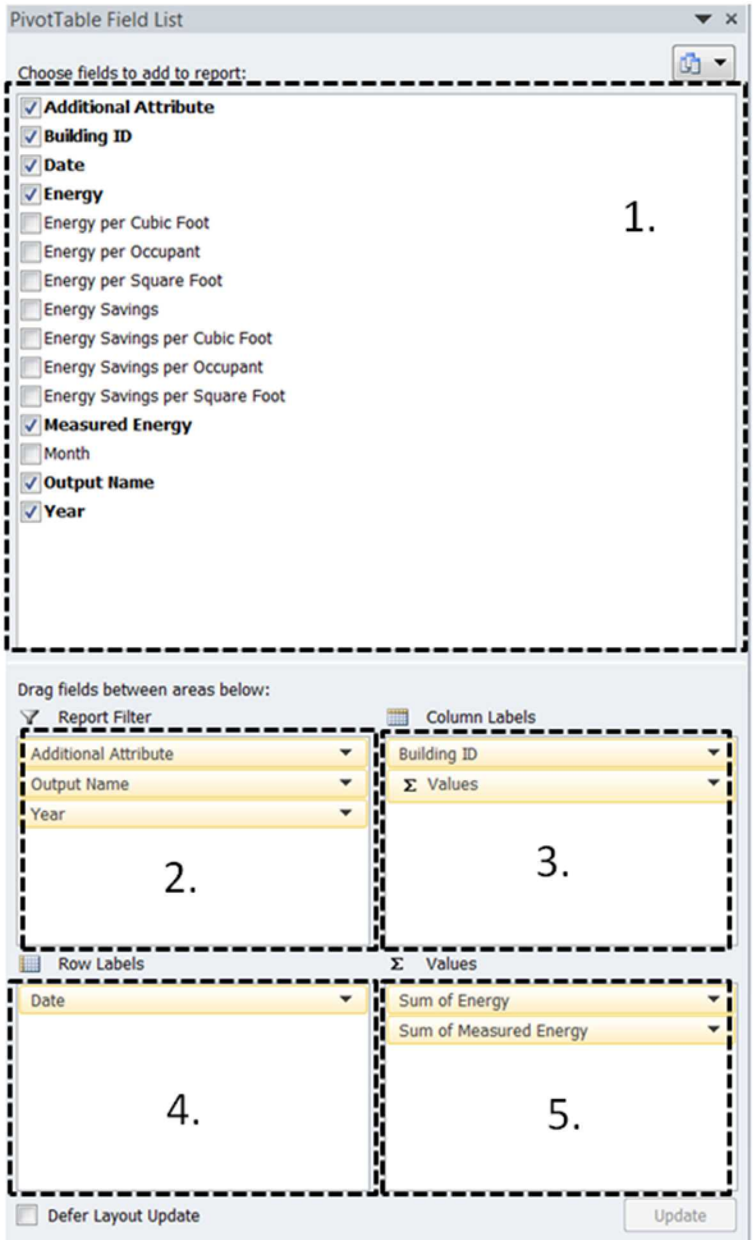


Figure 22. Drag and drop regions

Attribute fields can be assigned as report filters, column labels or row labels as seen in the lower portion of Figure 21. When an attribute is a report filter it does not affect the form of pivot table but allows data in the pivot cache to be included or excluded. When a report filter field is selected, the set of unique values attained by a field is displayed and the user can select one or more of the set as seen in Figure 23 for output name. For this example selecting "All" means that total electricity and gas are summed together, while selecting only one of them means that only electricity or only gas is displayed. An issue that can cause problems for this example exists if subcategories of energy uses are included. For example, if the user also included electricity due to HVAC equipment, then selecting total electricity and HVAC equipment double counts the electricity due to HVAC equipment because total electricity includes the HVAC equipment electricity.

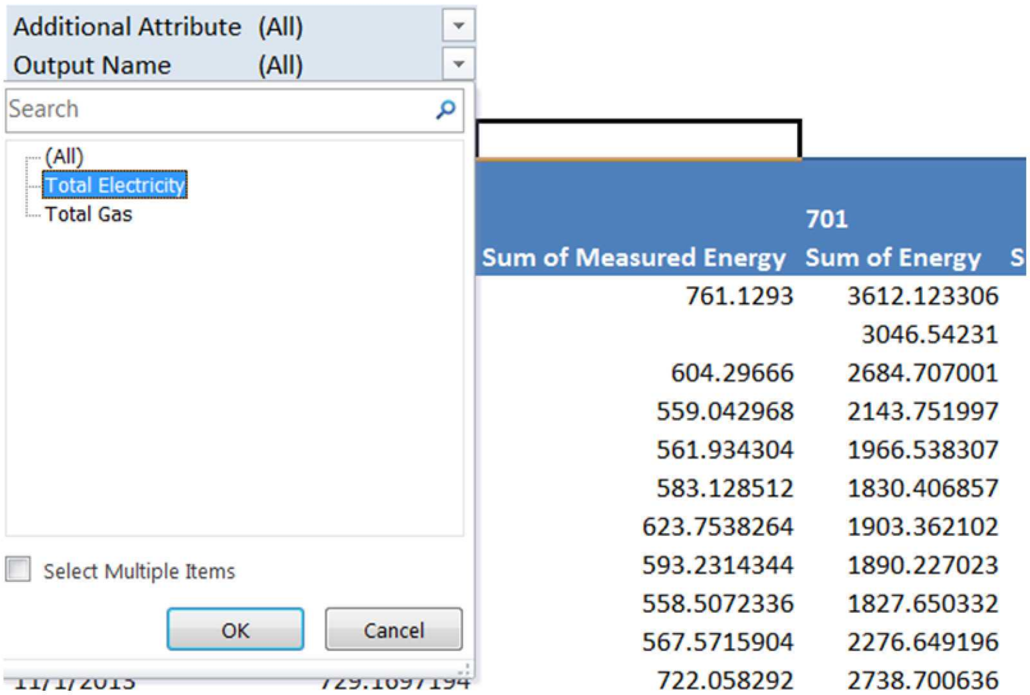
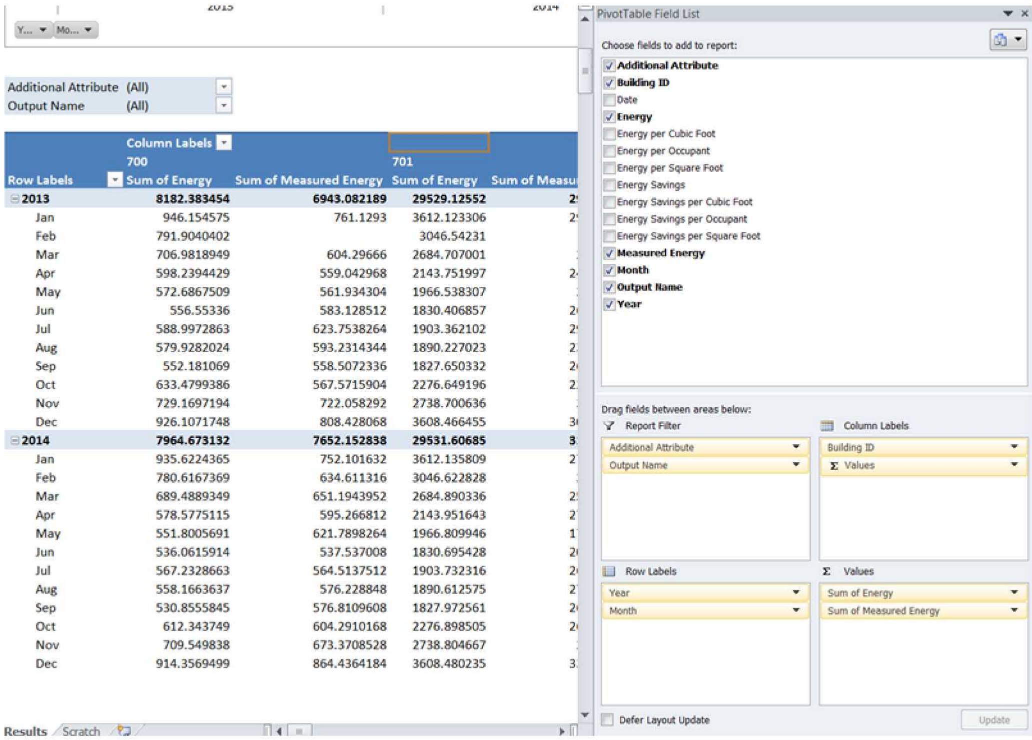


Figure 23. Output name as report filter

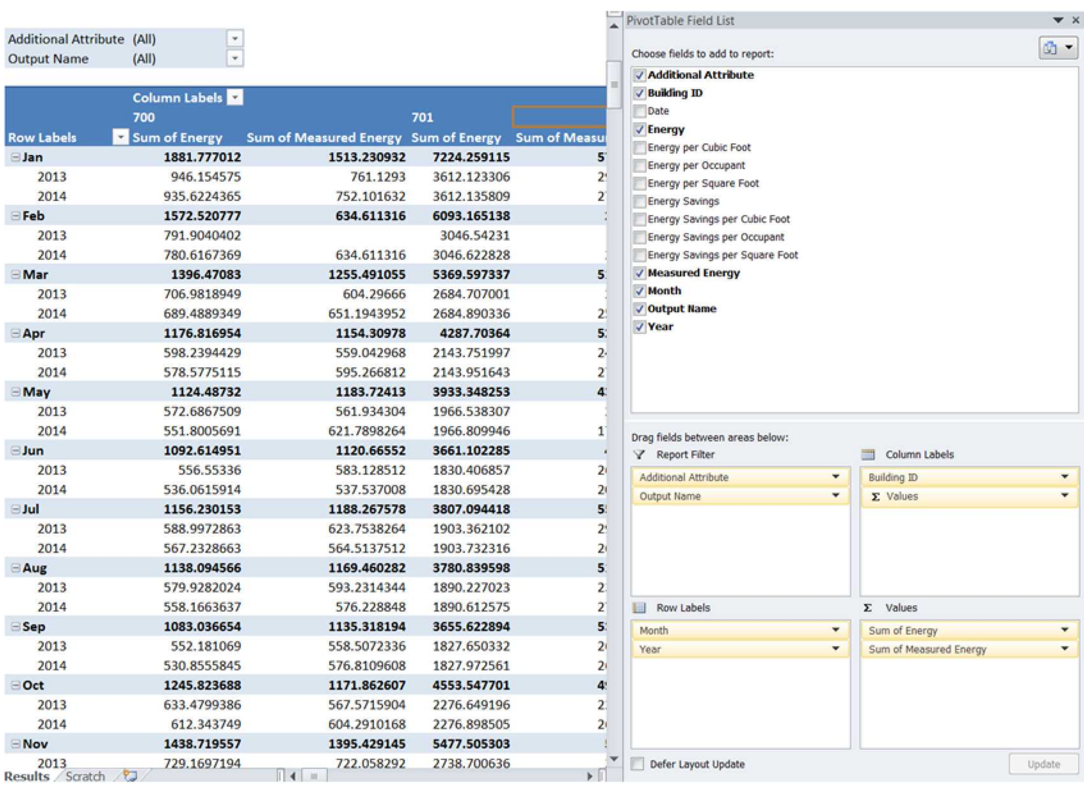
Row and column fields affect the form of the pivot table. Column fields display as the heading for columns, whereas rows fields display as the heading to rows. Several layers of row and column fields can be used to cause the table to sum over different fields. For example, if the year and month are both included as row fields, as seen in Figure 24, then this produces a pivot table giving yearly summation first, followed by monthly values. Swapping the month and year causes summation to be over the month of each year as seen in Figure 25. Alternatively, the summation can be turned off by navigating to field settings as seen in Figure 26.

Value fields usually consist of numeric data. Every entry in the pivot cache that contains the same attribute values as defined in the pivot table is represented by a single value in the pivot table where all rows that apply in the pivot cache are reduced to a single value through a function. The most common function is summation. Other functions include count, minimum, maximum, average, and standard deviation.



Row Labels	700 Sum of Energy	700 Sum of Measured Energy	701 Sum of Energy
2013	**8182.383454**	**6943.082189**	**29529.12552**
Jan	946.154575	761.1293	3612.123306
Feb	791.9040402		3046.54231
Mar	706.9818949	604.29666	2684.707001
Apr	598.2394429	559.042968	2143.751997
May	572.6867509	561.934304	1966.538307
Jun	556.55336	583.128512	1830.406857
Jul	588.9972863	623.7538264	1903.362102
Aug	579.9282024	593.2314344	1890.227023
Sep	552.181069	558.5072336	1827.650332
Oct	633.4799386	567.5715904	2276.649196
Nov	729.1697194	722.058292	2738.700636
Dec	926.1071748	808.428068	3608.466455
2014	**7964.673132**	**7652.152838**	**29531.60685**
Jan	935.6224365	752.101632	3612.135809
Feb	780.6167369	634.611316	3046.622828
Mar	689.4889349	651.1943952	2684.890336
Apr	578.5775115	595.266812	2143.951643
May	551.8005691	621.7898264	1966.809946
Jun	536.0615914	537.537008	1830.695428
Jul	567.2328663	564.5137512	1903.732316
Aug	558.1663637	576.228848	1890.612575
Sep	530.8555845	576.8109608	1827.972561
Oct	612.343749	604.2910168	2276.898505
Nov	709.549838	673.3708528	2738.804667
Dec	914.3569499	864.4364184	3608.480235

Figure 24. Two layers of row fields



Row Labels	700 Sum of Energy	700 Sum of Measured Energy	701 Sum of Energy
Jan	**1881.777012**	**1513.230932**	**7224.259115**
2013	946.154575	761.1293	3612.123306
2014	935.6224365	752.101632	3612.135809
Feb	**1572.520777**	**634.611316**	**6093.165138**
2013	791.9040402		3046.54231
2014	780.6167369	634.611316	3046.622828
Mar	**1396.47083**	**1255.491055**	**5369.597337**
2013	706.9818949	604.29666	2684.707001
2014	689.4889349	651.1943952	2684.890336
Apr	**1176.816954**	**1154.30978**	**4287.70364**
2013	598.2394429	559.042968	2143.751997
2014	578.5775115	595.266812	2143.951643
May	**1124.48732**	**1183.72413**	**3933.348253**
2013	572.6867509	561.934304	1966.538307
2014	551.8005691	621.7898264	1966.809946
Jun	**1092.614951**	**1120.66552**	**3661.102285**
2013	556.55336	583.128512	1830.406857
2014	536.0615914	537.537008	1830.695428
Jul	**1156.230153**	**1188.267578**	**3807.094418**
2013	588.9972863	623.7538264	1903.362102
2014	567.2328663	564.5137512	1903.732316
Aug	**1138.094566**	**1169.460282**	**3780.839598**
2013	579.9282024	593.2314344	1890.227023
2014	558.1663637	576.228848	1890.612575
Sep	**1083.036654**	**1135.318194**	**3655.622894**
2013	552.181069	558.5072336	1827.650332
2014	530.8555845	576.8109608	1827.972561
Oct	**1245.823688**	**1171.862607**	**4553.547701**
2013	633.4799386	567.5715904	2276.649196
2014	612.343749	604.2910168	2276.898505
Nov	**1438.719557**	**1395.429145**	**5477.505303**
2013	729.1697194	722.058292	2738.700636

Figure 25. Flipped month and year pivot table

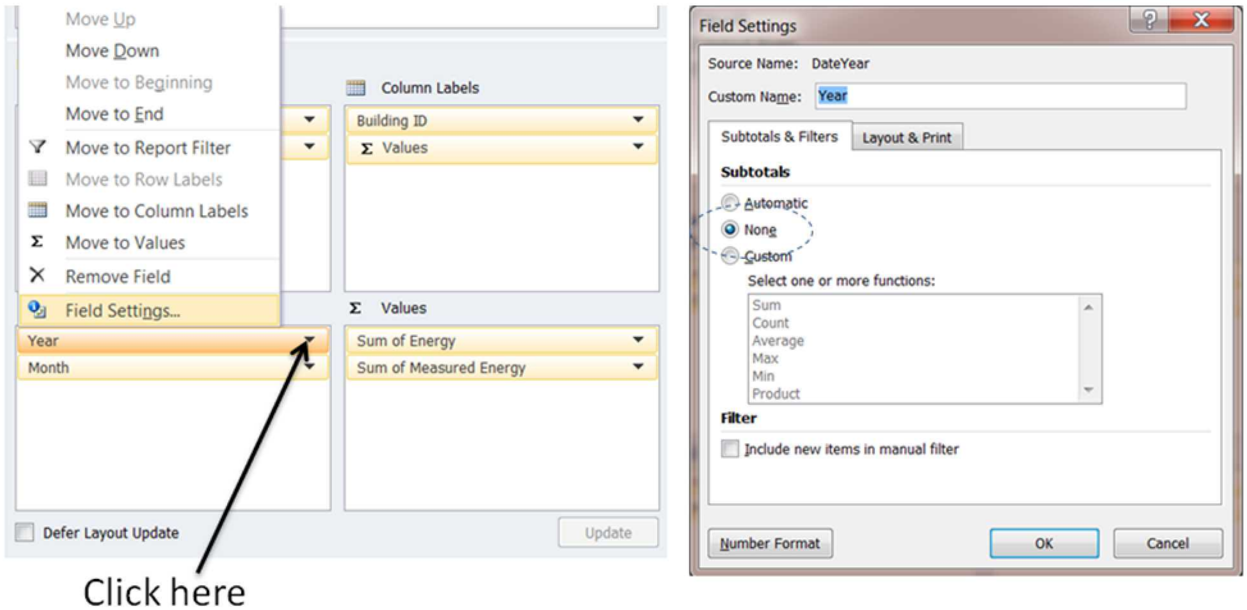


Figure 26. Changing field settings

Spending time working with pivot tables will reveal how useful they are as the user becomes more familiar with them. Wikipedia has a good article on pivot tables and many other tutorials are available on their use.

3 The IX Database

The IX database is a Microsoft Access® database containing all of the information generated by the IX user interface. Most users of IX should not have to do anything with the database besides occasionally compacting and repairing it. The IX database is a relational database with many relations between dozens of tables as seen in Figure 27. These relationships affect what kinds of operations can be accomplished. Add, edit, and delete operations may propagate between tables or could be forbidden because a field is subservient to another field in a different table. It is therefore very important that the interfaces described in the next sections be used instead of direct operations on the tables. The interfaces often cause VBA functionality to relate the tables in ways that are not conveyable through simple relations.

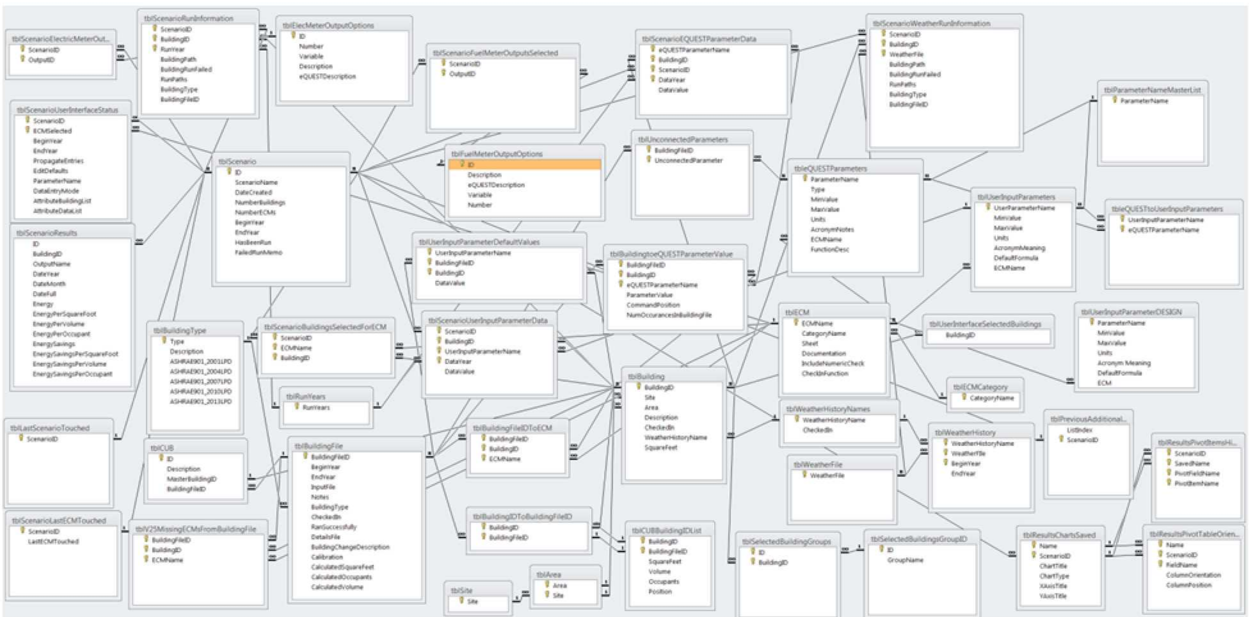


Figure 27. IX database table relationships

The main operations that have to be performed in the database are creation of ECMs, check in and check out for buildings, and check in and check out of weather histories. All of these functions are accessible through the main form. The main form has five command buttons that all open other forms as seen in Figure 28. Other operations that do not have an interface created are sometimes necessary. These operations have to be done with caution because they involve directly changing table records.

It should also be noted that the creation of new buildings, ECMs, and weather histories has received more attention than deletion and editing operations. There may still be inconsistencies in the database structure for these operations. For example, deleting an ECM may cause problems for checked in buildings that contain the ECM

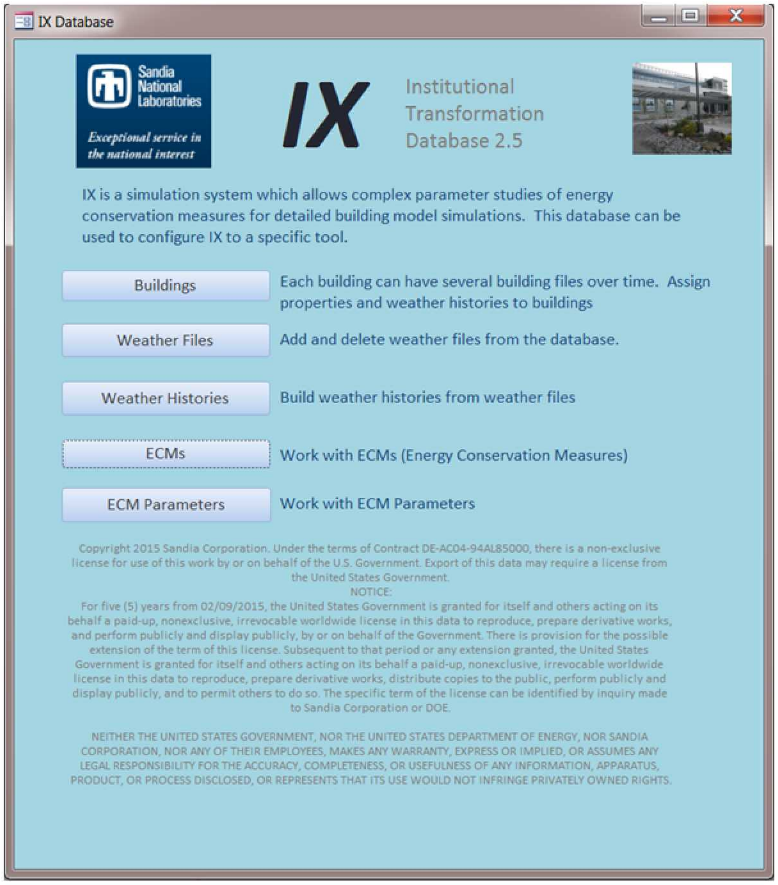


Figure 28. IX database main form

3.1 Common Functionality in Database Forms

Microsoft Access® forms usually show table records in the database. A record is a single row in a database table. Each row has the number of fields in the table. The user must understand that changes to a record are not completed until the next record is selected or the form is exited. Sometimes the user can spend quite a bit of time inputting data only to find that the entire record cannot be saved because one of the inputs is invalid. This can be frustrating because feedback occurs at the end of process and the message provided may be difficult to understand. Any record based form has common buttons that help the user navigate through records as seen in Figure 29.

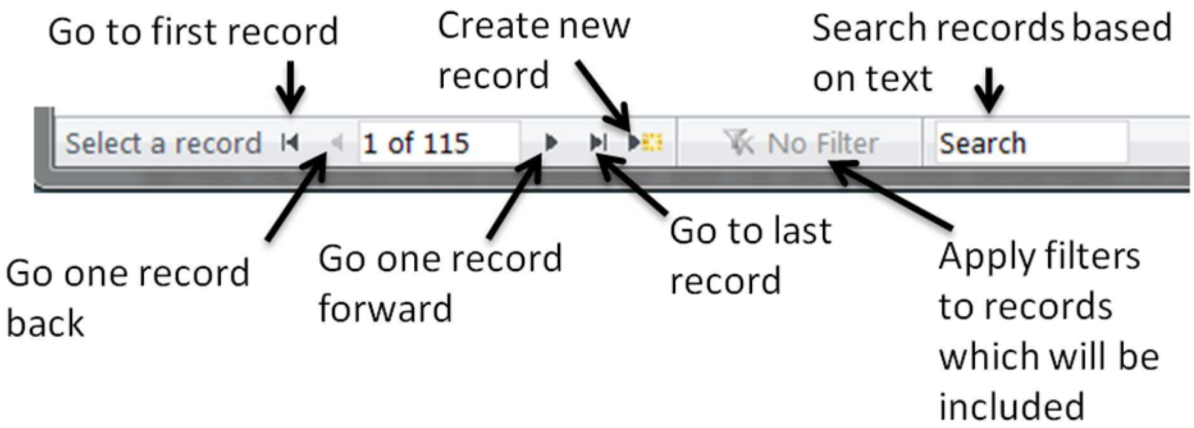


Figure 29. Database form record navigation

3.2 Buildings

When the "Buildings" button is clicked in the main form, the buildings form is opened as seen in Figure 30. The user can navigate to any building in the database through this form. No changes can be made to a building's input data until it is checked out. Checking a building out removes it from the list of eligible buildings for use in the IX user interface. It is therefore important to check a building back in after making changes. Each building is represented by one or more building files as seen in the "Building File List" list box. Double clicking on an entry in this list box opens up the building file form as seen in Figure 31. For buildings, there are two levels of checking in and out. The first level is at the building level. The user must first check out at the building level in order to check out at the building file level. Because of central utility buildings (CUBs), there is a many-to-many relationship between building files and buildings. A CUB building file can contain several buildings but a building can have several different building files for different years. As a result, it is necessary to check out every building associated with a building file before the building file can be checked out as seen for CUB 850 in Figure 32.

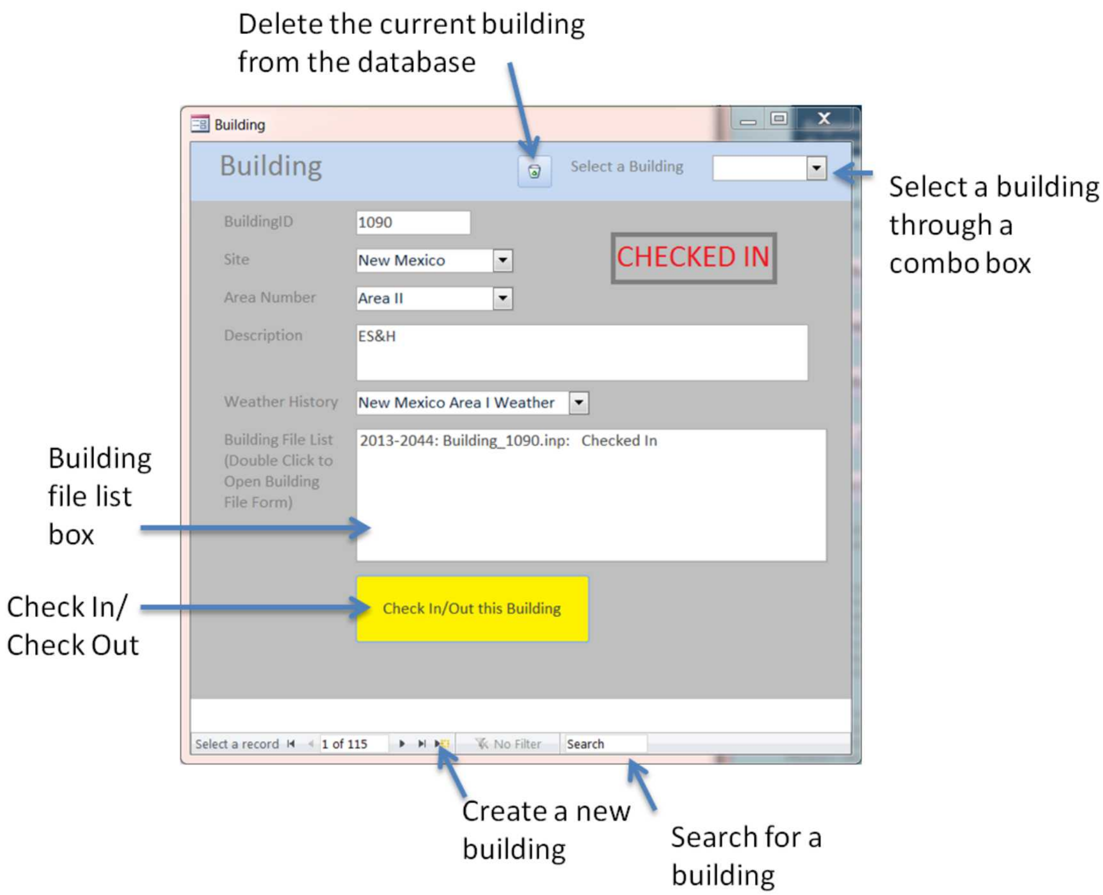


Figure 30 Building form

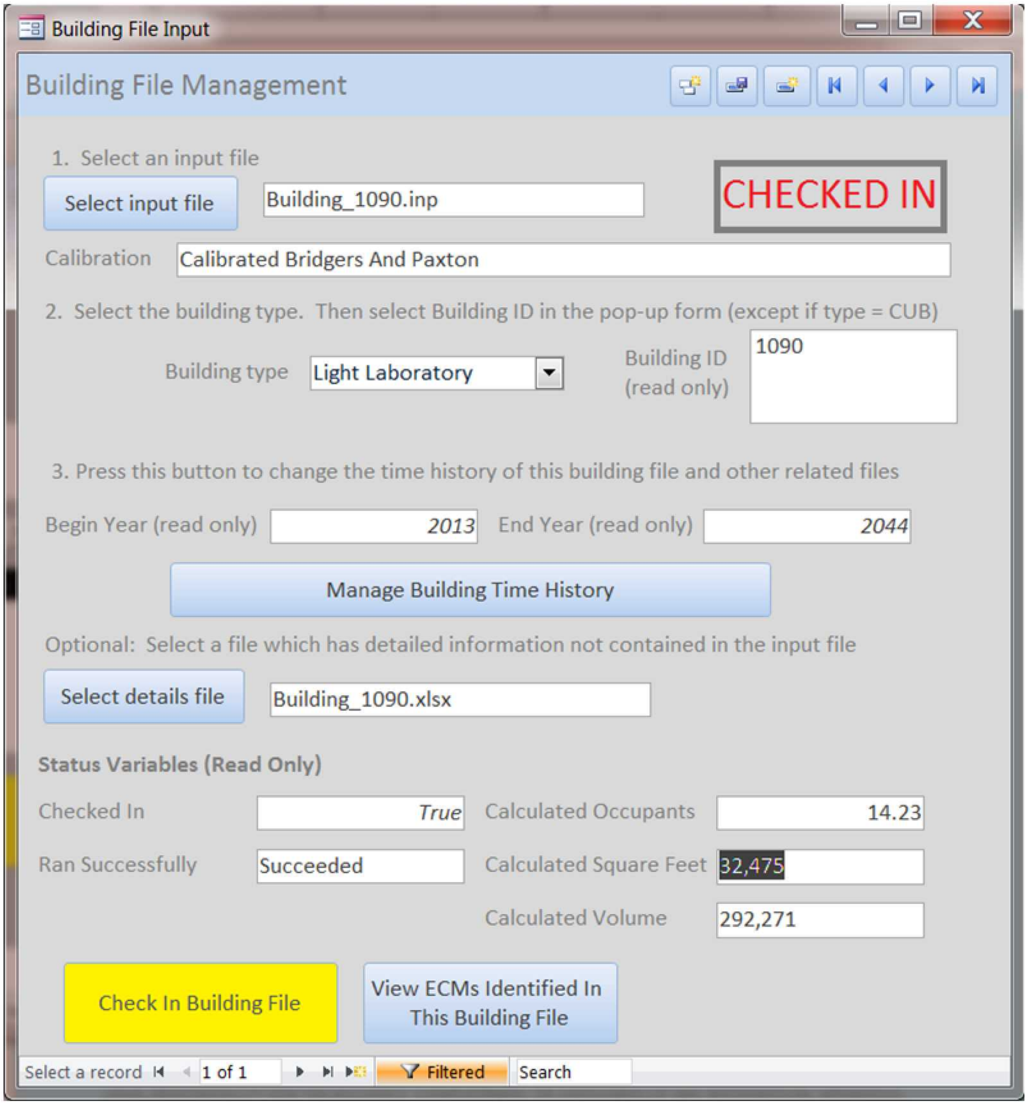


Figure 31. Building file form

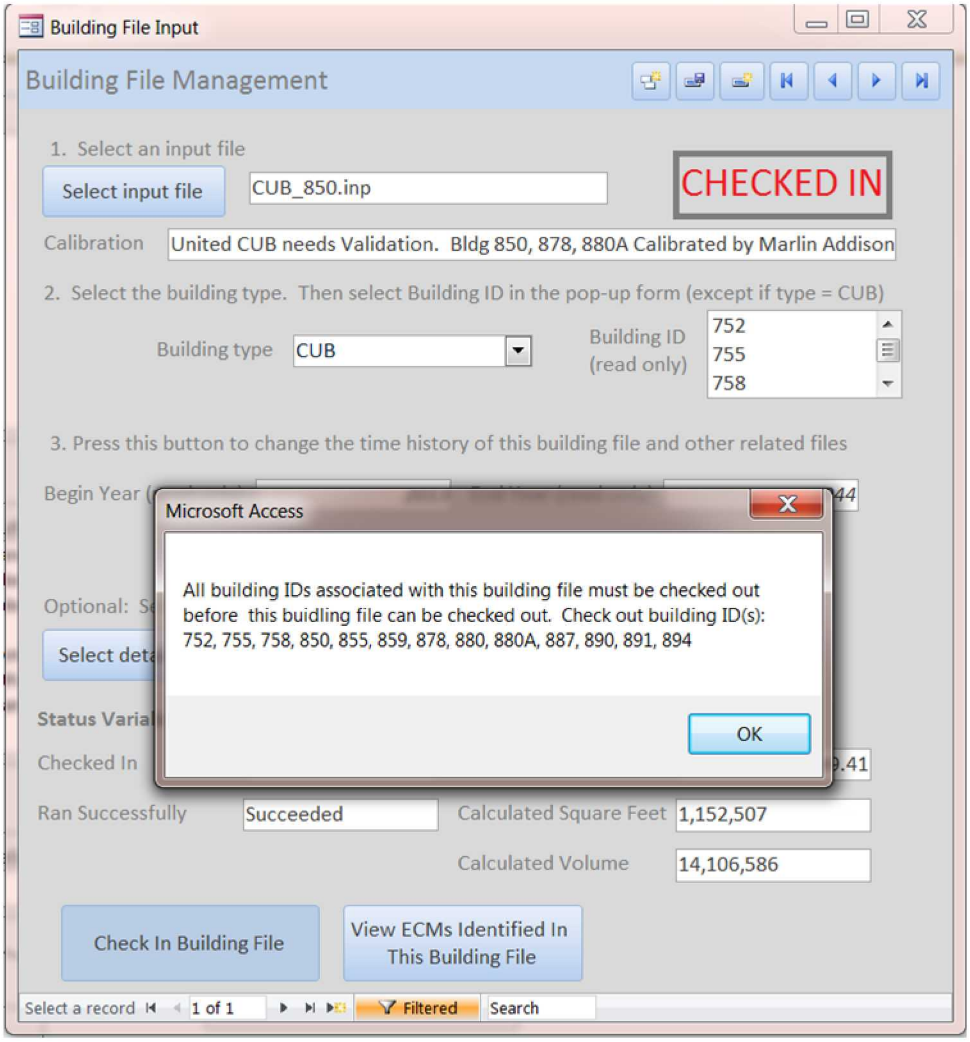


Figure 32. Associated buildings must be checked out first

Once buildings and building files have been checked out, changes can be made to a building file. Some of the changes that can be made are:

1. Change the file associated with the building file record by pressing the “Select input file” button. This is only recommended if a file has been found to be inaccurate. IX allows for older models to be retained for historical years so it is often a better course of action to create a new building file record.

2. Select the building type in the building type combo box. Changing this field has two reactions. If the building type is CUB, then a VBA routine looks for the following text at the beginning of the selected building input file:

   $ Begin Building List

   $ 858N,870,895,886,879,858S,857,703,702,701,700

   $ End Building List

The list shown here is for the 858N CUB building file. Without this list, a file cannot be assigned type CUB. If the type is anything else, then a form pops up which allows the user to select which building ID the building file is to be associated with.

3. Manage the building file ID time history by pressing the "Manage Building Time History" button. A form appears which allows the user to change the time history of which years apply to a given building file as seen in Figure 33. This is accomplished by clicking on an entry in the list box, selecting begin and end years, and then pressing "Define/Change Selected History."

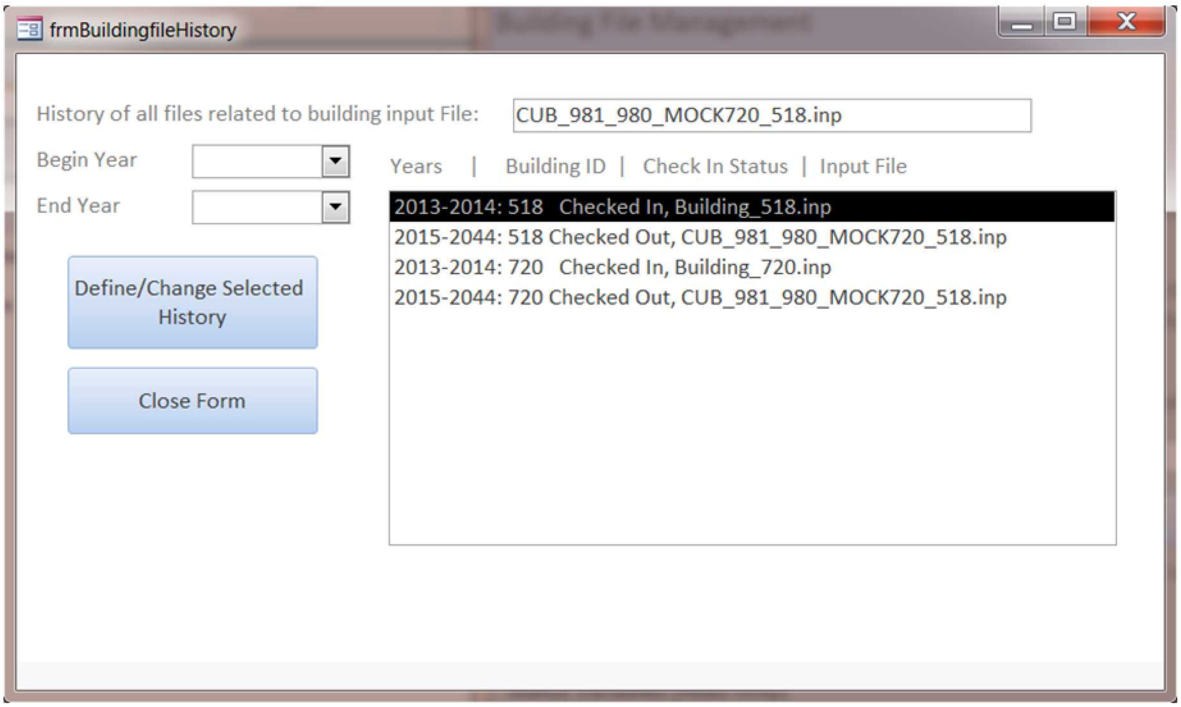


Figure 33. Managing building file time histories

It is up to the user to make sure that each year is only applied to one building file ID. The form can be closed with inconsistencies but the building file will not check in until a consistent history has been created as seen in Figure 34.

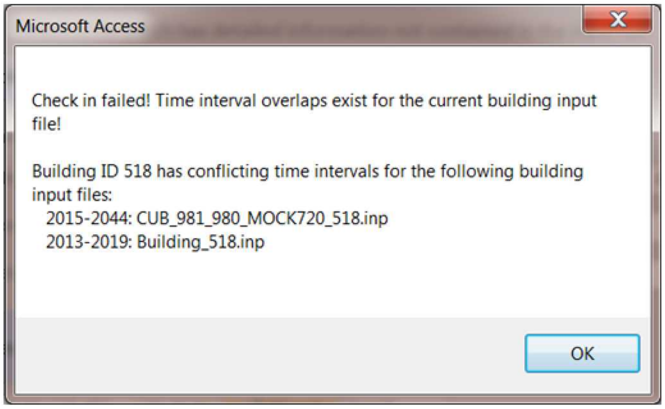


Figure 34. Check in failure due to time interval overlaps

4. Select a details file by pressing "Select details file." The details file is not critical to IX's functionality but is critical for informational purposes. It is expected that specification of the nature of calibration and of assumptions in the model will be included in this details file. The details files are moved to ./IXData/BuildingDetailedInformation.

3.3 Weather

Selecting the second and third buttons seen in Figure 28 enables the user to create weather histories. IX is confined to having a single weather file for each year. A weather history is a set of weather files assigned to different years. The first button "Weather Files" opens up a form allowing the user to add and edit weather files to the database as seen in Figure 35.

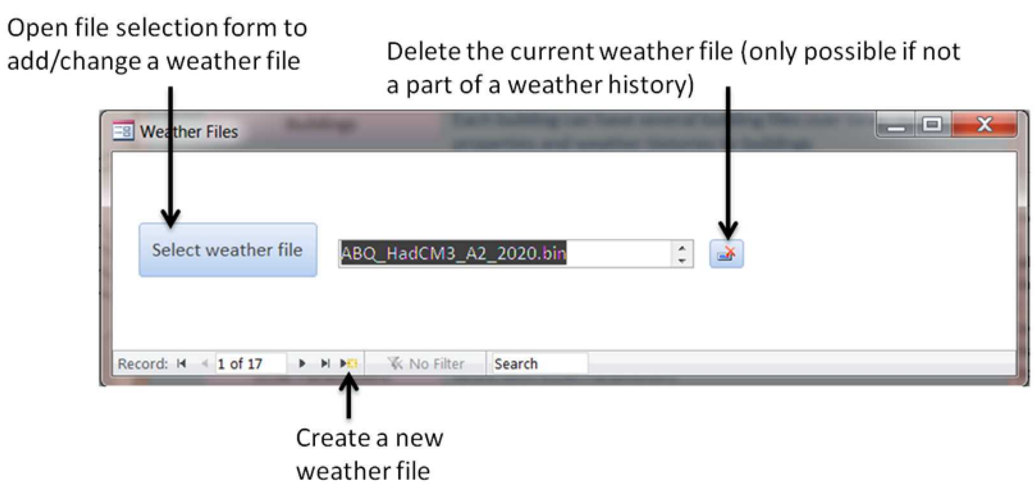


Figure 35. Weather file form

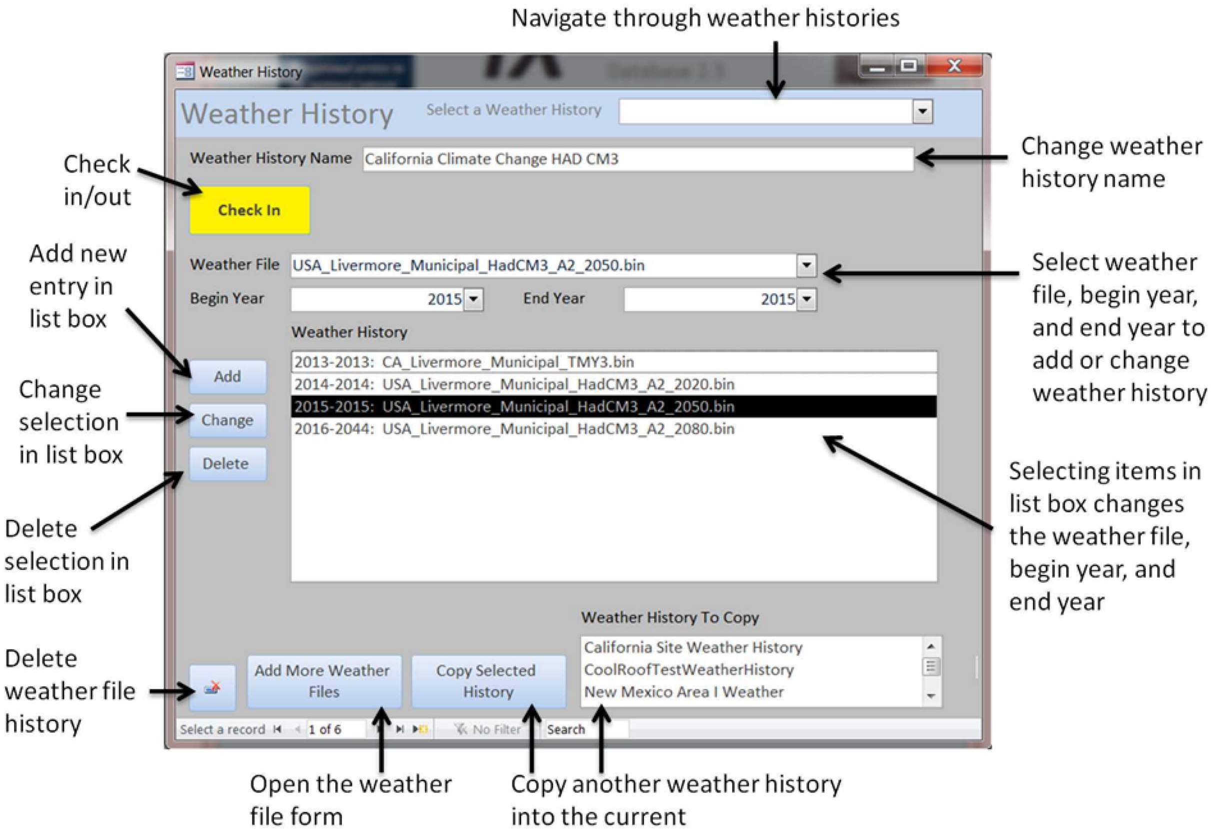


Figure 36. Weather history form

Weather histories have to be checked in and out. This is necessary because combinations of weather files and building files have to be run inDoe2.2 for a building to be checked in. It is therefore necessary for a weather history to be checked in before a building can use it. Once a building uses a weather history, the weather history cannot be changed until the building using it has been checked out.

3.4 Energy Conservation Measures

Energy conservation measures (ECMs) are the most difficult function to work with in the database. To create a new ECM, the user must first design the ECM. An ECM consists of two sets of parameters that have to be input into the database. The first set is called the user input parameters. These are the inputs that the user will have control of. The second set is called the eQUEST parameters. These are the parameters inside each building input file. Two functions have to be designed between the two sets. The first is a function that maps the user input parameters to the eQUEST parameters and the second is a function that maps the reverse direction—eQUEST parameters to user input parameters. This function can be as simple as a one-to-one mapping both ways or as complex as a two custom VBA functions.

The next step is to create an empty ECM by pressing the “ECMs” button from the main form shown in Figure 28. This opens the form seen in Figure 37. Once an ECM has been added, it can be accessed through the “ECM Parameters” button, which opens the form seen in Figure 38. After navigating to the ECM just created, the ECM’s content is created in this form using seven steps.

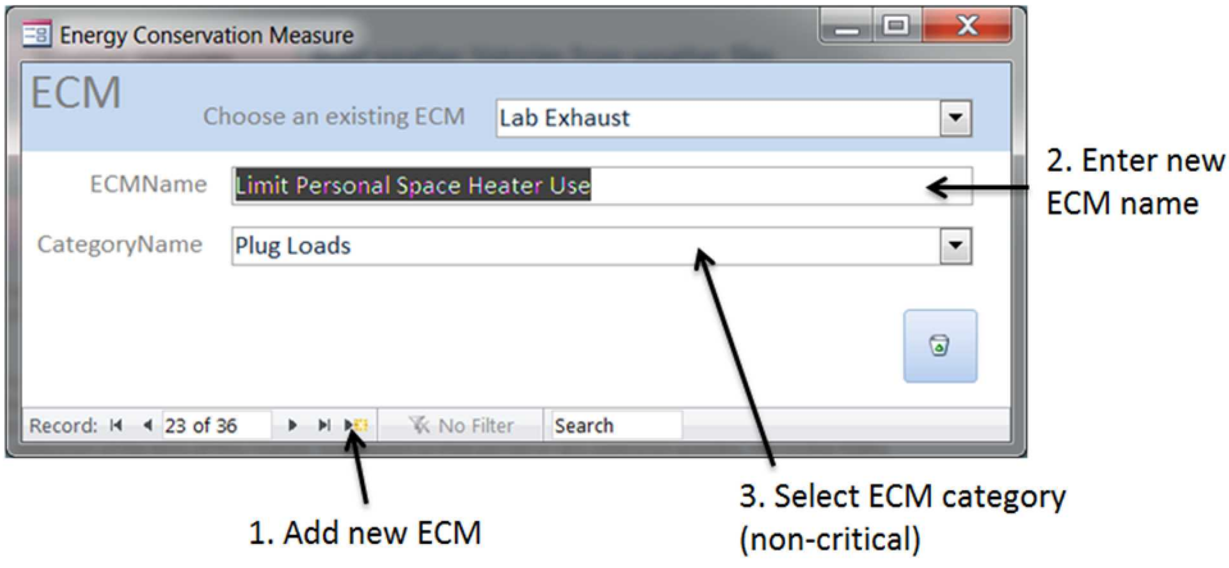


Figure 37. ECM form

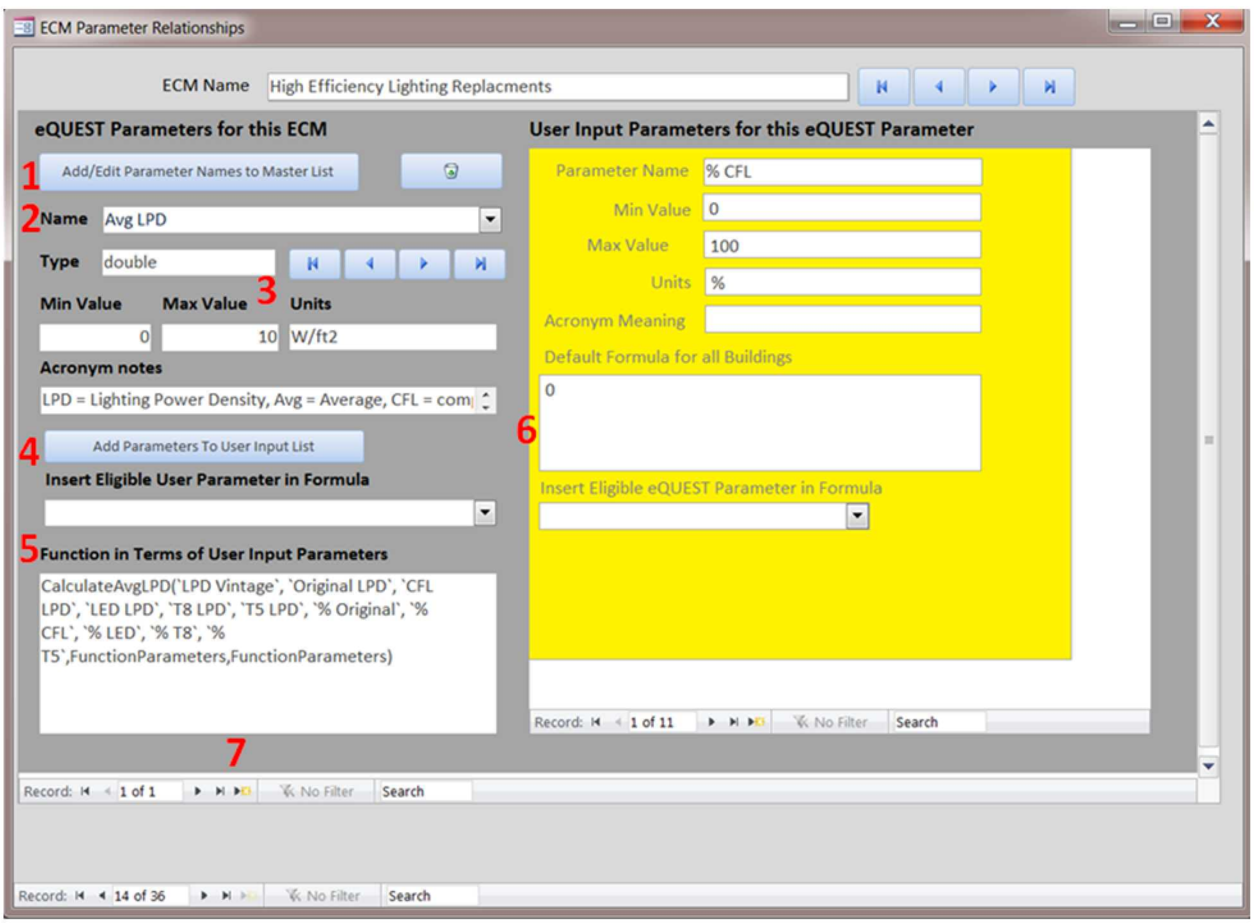


Figure 38. ECM parameters form

1. Add all parameter names for the ECM for both user input parameters and for eQUEST parameters by pressing "Add/Edit Parameter Names to Master List." This opens up another form allowing the addition of parameters as seen in Figure 39. The new parameters will be selected in steps 2 and 5. Be extra careful to match names that are in any building input files because renaming can be difficult.

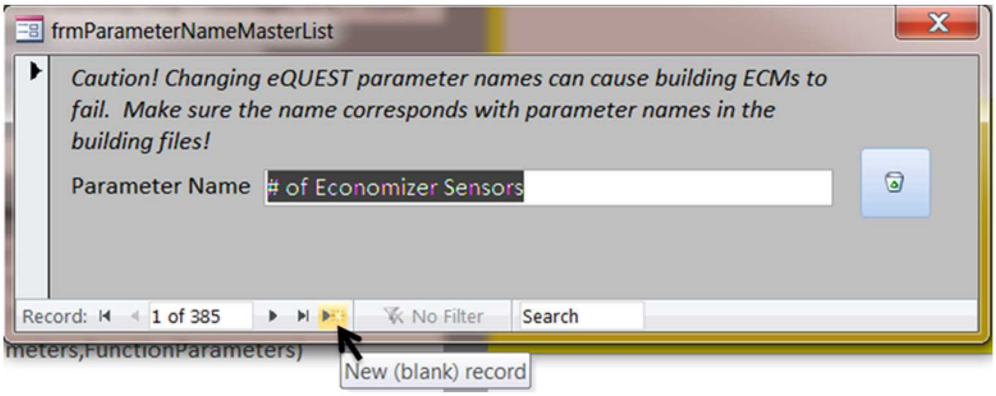


Figure 39. Parameter master list form

2. Select the first eQUEST parameter name from the "Name" combo box. Only parameter names that are unused by other ECMs are included in this list.

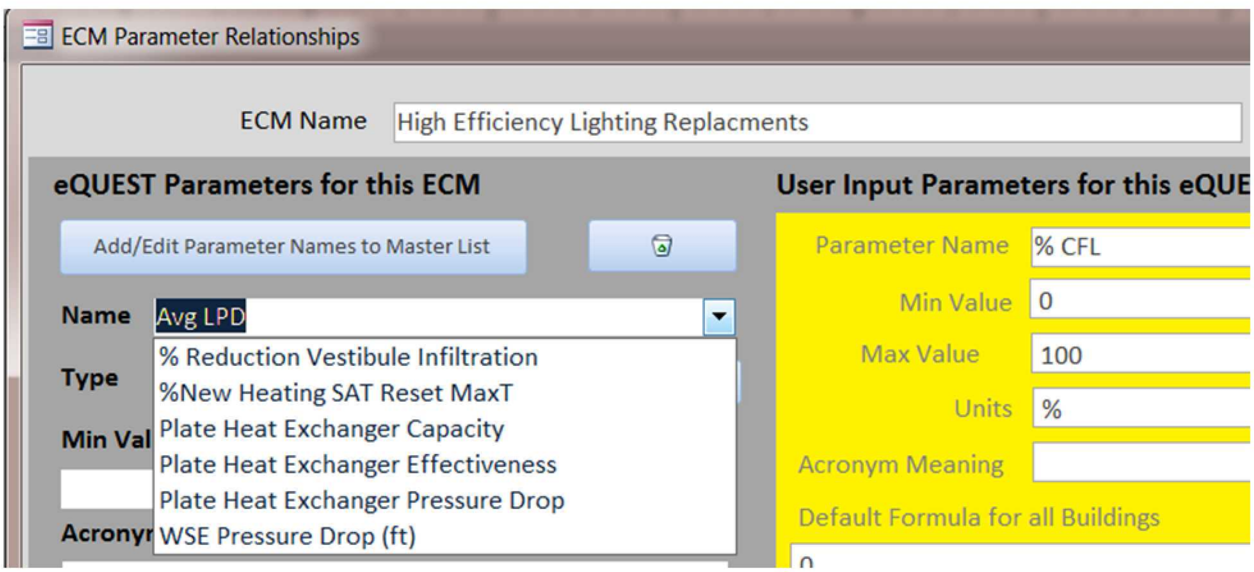


Figure 40. Select from combo box

3. Enter values for type of variable (non-critical), Min value, Max value, Units, and Acronym notes. The acronym notes field exists to specify the meaning of any abbreviations or acronyms used in the eQUEST parameter name

4. Press "Add Parameters to User Input List" to define which of the parameters in step 1 are user input parameters. This opens up a new form, as seen in Figure 41, which allows the user to take parameters from the master parameter list and add them to be eligible to be user input parameters.

They are not added as actual user input parameters until they are included in a function to calculate the eQUEST parameters. Five steps are necessary to process each parameter:

a. Create a new record
b. Select the ECM currently being worked on
c. Select one of the parameter names created in step 1
d. Enter Min Value, Max Value, Units, and Acronym Meaning
e. Repeat a through d or close form if all parameters have been processed

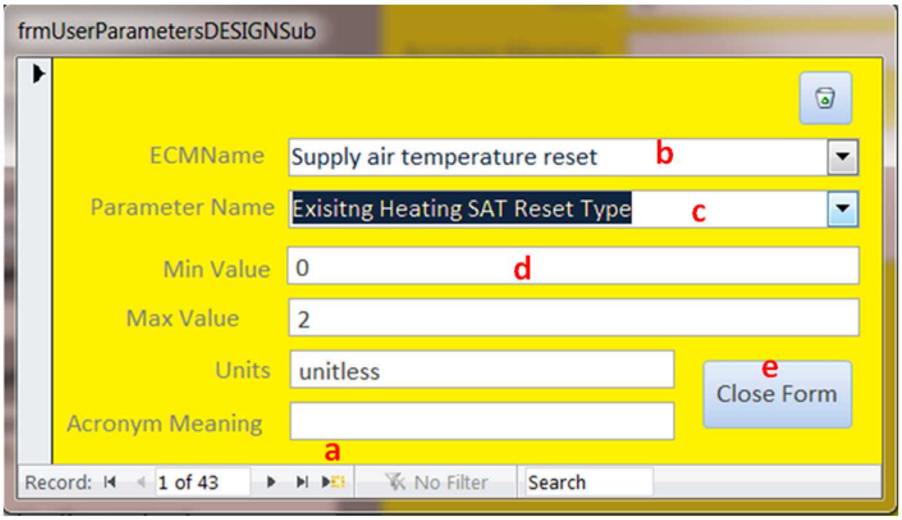


Figure 41. Add master list parameters to user input parameter list

5. Enter a function of the user input parameters that determines the current eQUEST parameter. This can be a call to a VBA function or an arithmetic combination of the user input parameters. Parameter names must be enclosed with the "`" character (Figure 42). Selecting the "Insert Eligible User Parameter in Formula" combo box gives the list of parameters that have been added in step 4 and that enters one of the names in the "Function In Terms of User Input Parameters" text box automatically. At this point the user must understand that there is a special VBA subroutine that is executed every time the "Function In Terms of User Input Parameters" text box is exited. This subroutine verifies that the text box input correctly evaluates to a real number and adds records to relate all of the user input parameters contained in the text box input. Because of this subroutine, once input has been started, the user must keep that formula consistent every time that the text box is exited. Otherwise an error will result as seen in Figure 43 where a "+" has been placed at the end of the formula without anything on its right hand side. If several parameter names need to be added, it is best to add them one by one while placing "+" between them so that a consistent formula is generated. The user can then update the formula to the desired form afterward. Once a valid function has been entered, the User Input Parameters in the function are available in the yellow portion of the ECM Parameter Relationships form (Figure 38).

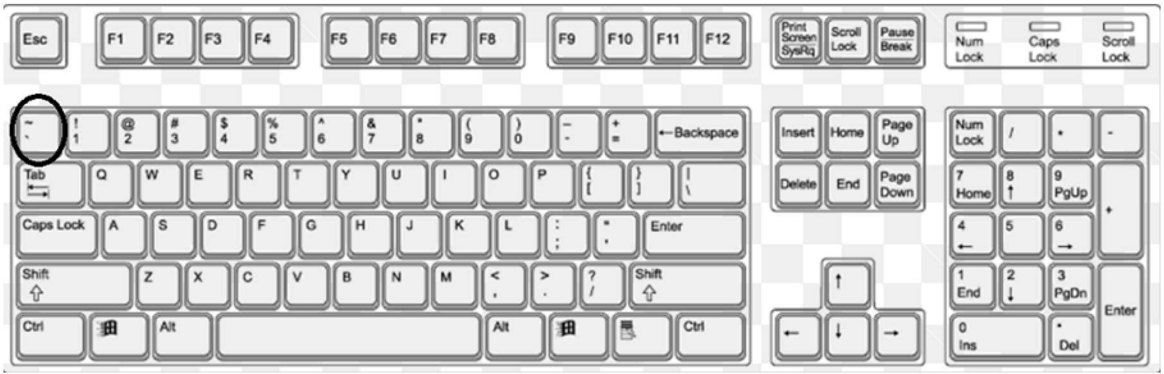


Figure 42. Character to delimit parameter names

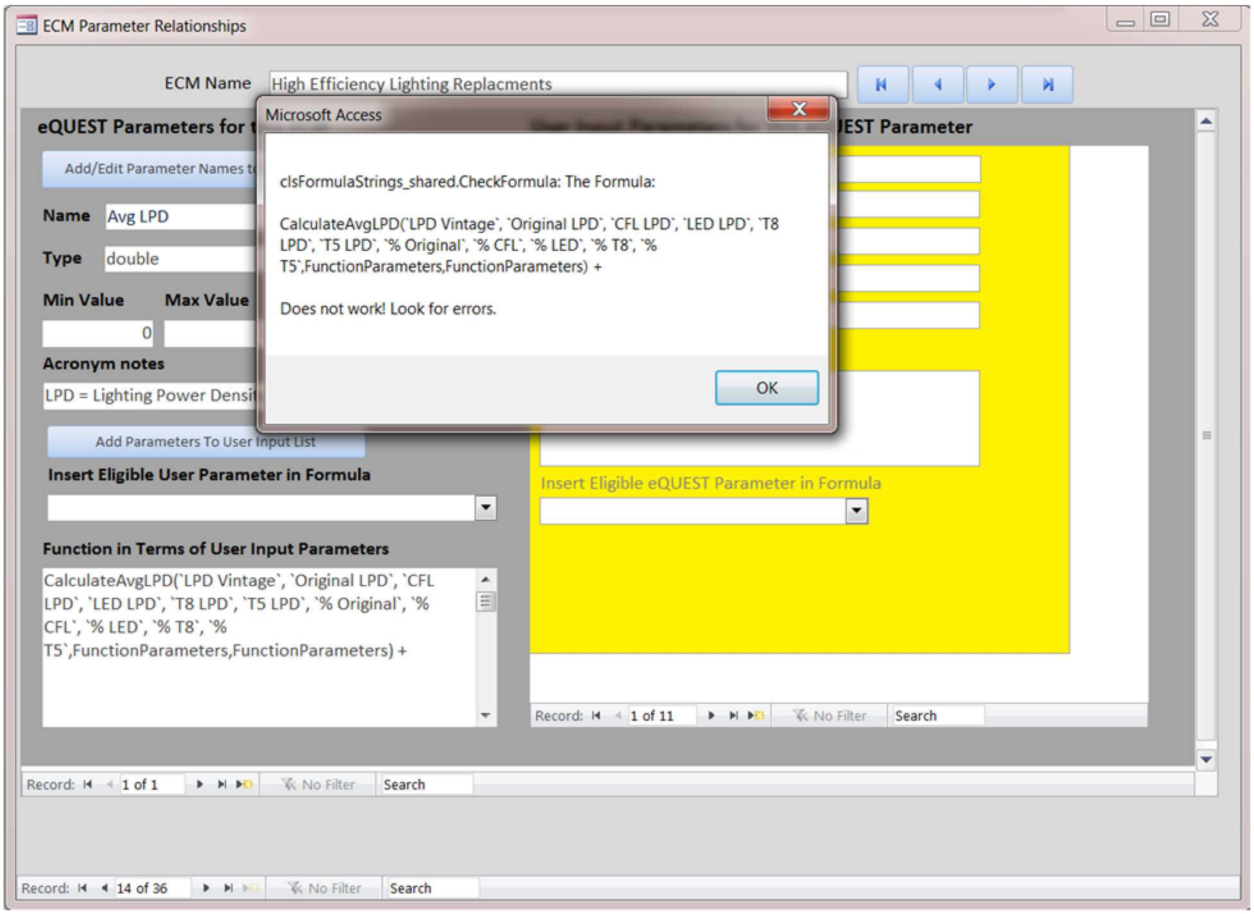


Figure 43. Inconsistent formula

6. For every user input parameter referenced by the function entered in step 5, enter a second function in the "Default Formula for All Buildings" text box which determines each user input parameter as a function of the eQUEST parameters. All the same principles and cautions in step 5 apply here. This second function is necessary because the IX database needs to be able to provide default start values of the user input parameters. For many cases this function is just the inverse

function of the function entered in step 5. It can be more complicated if more user parameters exist than eQUEST parameters.

7. If more eQUEST parameters than one exist in the ECM, then add a new record and repeat steps 2, 3, 5, and 6 (all parameters should be processed in steps 1 and 4) until all eQUEST parameters have been assigned

Once this process is complete, it is up to the user to add the necessary eQUEST parameters and expressions containing these parameters in the correct locations in each building input file. The building input files have to be checked out and then checked back in for the ECM to be added.

4 CUB Utility

The centralized utility buildings (CUB) Utility is a tool used to join building models into a single model so that they can share common chilled water loops. The utility accomplishes a large amount of the work needed to join the files but does not complete the job. After the files have been joined there is a procedure outlined at the end of this section that must be followed to make the final CUB file.

The CUB Utility can be found in .\IXData\CUB-BuildUtility\IXCUB-BuildUtility.xlsm. Before running the utility the user must place all of the individual building models to be joined together in the folder .\IXData\CUB-BuildUtility\OriginalFiles. There are two steps to run the utility (both take a while to perform). The first alters the individual files so that they have unique names, adds meters and applies these meters to the appropriate systems, and offsets all of the geometry so that buildings will have an appealing spatial lay-out. The second step joins all of the files and deletes out command blocks that can occur only once or only need to be defined in the master CUB file.

The utility's input screen is shown in Figure 44. The steps to use the utility are as follows:

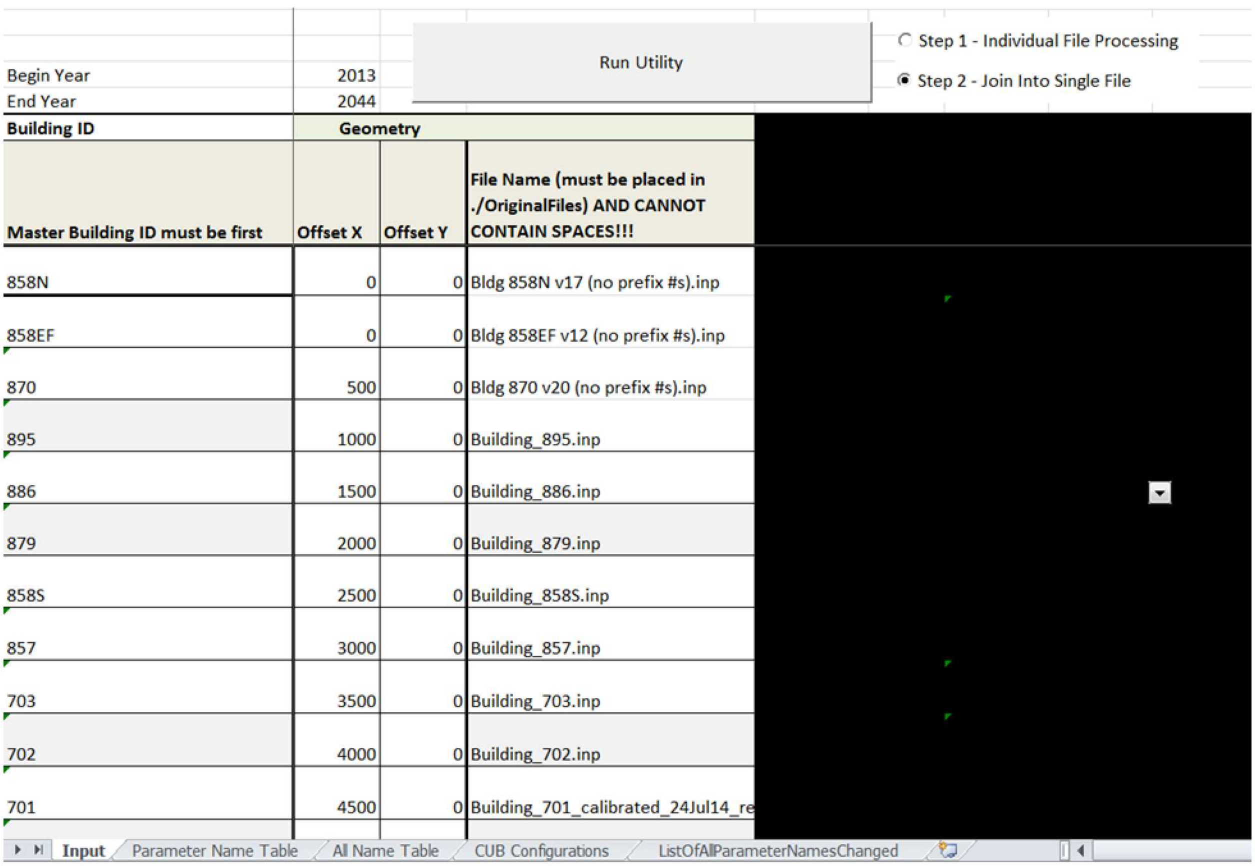


Figure 44. CUB utility input screen

1. Fill in building ID, offset X, offset Y and FileName (starting in cell A8) for all buildings. The first building must be the CUB building model. The file name must exactly match a file name in the ./OriginalFiles folder.

2. Enter a begin and end year for the CUB in cells B4-5 the only important part about this step is that the begin and end year are significant in data that has to be added to the database manually.
3. Select “Step 1” in the radio button control to the right of the “Run Utility” command button.
4. Press the “Run Utility” button and wait for the progress meter to complete.
5. Select “Step 2” and press the “Run Utility” button again.
6. In eQUEST, manually process the joined file which should be in the base directory .\IXData\CUB-BuildUtility\ and named “CUB_<CUBID>_<BID#1>_...<BID#n>” (BID = Building ID)

The manual process includes the following steps:

1. All building-level chilled water (CHW) Loops are converted from PRIMARY to SECONDARY and they are assigned to the CUB primary loop. If this is done in the Detailed Interface of eQUEST (recommended), recall that a building design language (BDL) error with a misleading error dialog appears on the screen after you switch the Loop-Type from Primary to Secondary. Click to dismiss the errant dialog. If close the Loop properties dialog and then re-select the building-level CHW loop you just converted from Primary to secondary, you will then receive the proper required keyword dialog prompting you for the CUB primary CHW loop.
2. Any chillers that previously served the building-level CHW loop (now secondary) should be deleted.
3. Any CHW loops that previously served the building-level CHW loop (now secondary) MAY need to be deleted if there are no building-level CHW pumps.
4. Any CW loops and cooling towers that previously served the building-level chiller should be deleted. Delete the cooling tower first, then the CW loop (this will delete the CW pump with the CW loop).
5. Delete and boilers and hot water (HW) pumps as appropriate.
6. Delete all building-level working system equivalent(WSE) components.
7. Delete parameters orphaned by the deletions above, e.g., building-level “…Chiller kW/ton”, “…Boiler Thermal Eff”, “WSE Capacity (gpm)”, ”0898WSE Pressure Drop (ft)”, “0898WSE Effectiveness”, etc.
8. Any building-level CHW or HW pumps and boilers that remain in the combined CUB model must be assigned to be served by the respective building-level electric meter (EM) 1 and fuel meter (FM) 1 meters. Confirm that the only equipment assigned to the Master Meters (EM2 and FM2) are the following CUB-level items: pumps, chillers, boilers (if any), and cooling towers. All building-level air handling units (AHUs) should have already been assigned to their respective building-level meters (-EM1 and -FM1) by Daniel Villa. This will default the lights, plugs etc. in

each building to be assigned to their respective –EM1 and –FM1 meters. The CUB building AHUs, lights, plugs, etc. will also be treated as building-level loads, i.e., for the CUB building.

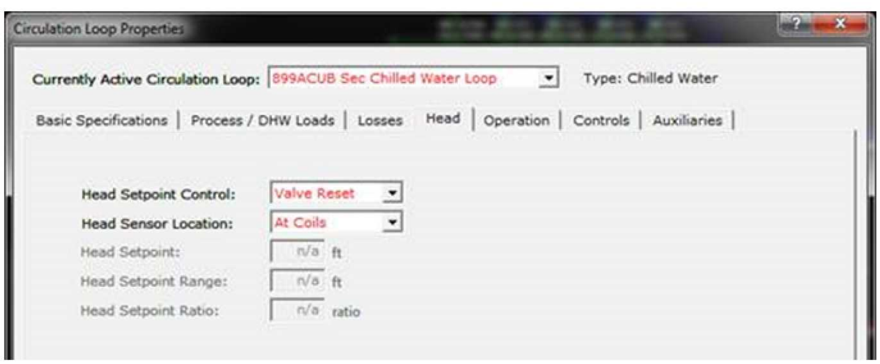
9. When Daniel transmits his 95% assembled CUB models he will include several rows of comments at the top of the INP file such as the following:
   $ Begin Building List
   $ 899A,898,899
   $ End Building List

   $ Begin Building Data
   $
   6935.4,10.2891802386757,117901.8,173370.8471,162.080094929273,3078229.1679,61731.067,62.1030606375946,1088220.377075
   $ End Building Data
   These comments lines MUST be pasted back into the final (calibrated) CUB model INP file (at the top) before sending the calibrated CUB model files back to him.

10. Check that the valve-type for each AHU in the CUB-served buildings are set correctly, i.e., 2-way or 3-way. Because the secondary (building-level) loops are always variable flow (with VSD on the CUB pumps), most of the valves will be two-way.

11. Use best judgment to assign the location of the various piping pressure drops, e.g., most will be at the secondary loop, confirm pressure drops (i.e., from plans) at coils and at the primary equipment (chillers, boilers, towers) and primary loops (CHW, CW, HW).

12. For the CUB primary CHW loop (and HW loop if any), set Head Set-point Control = Valve Reset and Head Sensor Location = At Coils (see screen capture below).



13. Play with various/appropriate Equipment-Control and Load-Management commands to replicate our guestimate regarding chiller (and boiler) staging. For 899A CUB, I assumed a OA temperature-based Load-Management command where the first stage was <55F and led with the WSE

Once the CUB is in the IX database, the Sheet "Parameter Name Table" must be imported into the IX database so that IX will be able to recognize parameter names from ECMs that have become longer than 32 characters and were assigned a new name in the CUB. Import to table "tblCUBParameterNameChanges." If this step is not completed some ECMs that are in the CUB file may not be included. This is necessary because each parameter name has a building ID added to it to keep parameter names unique for each building in the CUB file.

Appendix A ECM Descriptions

Air Management in IT Rooms and DCs

ECM Description This ECM seeks to improve building energy performance by controlling supply air characteristics (temperature, flow rate, etc.) to information technology (IT) rooms (server rooms, computer rooms, etc.) and data centers (DC) buildings to minimize cooling loads on HVAC equipment servicing these space/building types, thereby reducing HVAC system energy consumption. Revised suggested wording: When IT racks are free standing in IT rooms without physical containment (e.g., permanent partitions or plastic curtain barriers) the warm air exhausted from IT racks mixes with cool supply air intended to cool the IT equipment. **This ECM seeks to use IT rack equipment cooling capacity more efficiently by improving IT rack cooling air containment, i.e., reducing the amount of cooling supply air that mixes with IT rack 'exhaust' air**. The key parameter for this ECM is "IT AirMang Return Frac".

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
IT Air Management Return Fraction	0	1	Fraction	A large number (e.g., 0.75) implies a significant amount of mixing. A small number, e.g., 0.1, implies very good containment

Parameters In eQUEST/doe2.2

Name	Low Value	High Value	Units
IT AirMang Return Frac	0	1	Fraction

Functional Relationship

`IT AirMang Return Frac` = `IT Air Management Return Fraction`

How to implement the ECM in a new building:

This ECM is applied to every space which contains a significant amount of IT equipment which from the SPACE command requires input of "IT AirMang Return Frac" to an EQUIPMENT-KW keyword. An expression is required in this space using the parameter

EQUIPMENT-KW = ({**X***(1-#pa("IT AirMang Return Frac"))})

Where X is the total IT Equipment load in KiloWatts. The corresponding PLENUM to each space must have the following inserted.

EQUIPMENT-KW = ({**X***(#pa("IT AirMang Return Frac"))})

The underlying assumption of the ECM is that when mixing occurs, the heat load is directed to the plenum. Otherwise the heat load stays in the original space.

Airside Economizer Control

ECM Description: This ECM seeks to improve building energy performance by applying control of the quantity of outside air entering the building through an outside air damper to maximize the ability to supplement or replace HVAC system-produced conditioned air with outside air that already meets the conditioned air requirements (temperature and humidity), thereby reducing energy demands on the building chiller/boiler systems when the appropriate outside air conditions exist (temperature and humidity).

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
# of Economizer Sensors	0	2	none	0 = No sensors – a fixed amount of air will be let in whenever fans are running regardless of the "Economizer Type" 1 = One sensor is used and air is only let in when the sensor reading is below a constant maximum allowable value Note: The maximum allowable value is not controlled by this ECM and is controlled in BDL by the DRYBULB-LIMIT (default value is 65F) or ENTHALPY LIMIT (default value is 30.0 BTU/lb) keywords) 2 = Two sensors are used and air is let in when the outside sensor reading is less than the return air reading
Economizer Type	1	2	none	1 = temperature based economizer 2 = enthalpy based economizer
Economizer Upper Temperature Limit	45	99	Fahrenheit	Doe2.2 manual does not describe this input. It changes the ECONO-LIMIT-T keyword in a SYSTEMS command

Maximum Percent Outside Air	0	100	%	The maximum amount of outside air allowed into the building.
Minimum Percent Outside Air	0	100	%	The minimum amount of outside air allowed into the building.

Parameters In eQUEST/doe2.2 (read only)

Name	Low Value	High Value	Units	Explanation
Econo Type	0	4	none	0 = No sensors FIXED amount of outside air 1 = Use outside air temperature and a fixed temperature upper limit to determine when to let outside air in 2 = Use outside air enthalpy and a fixed enthalpy upper limit to determine when to let outside air in 3 = Let in air when outside air temperature is less than return air temperature 4 = Let in air when outside air enthalpy is less than return air enthalpy
Econo UpLimit T	45	99	Fahrenheit	=Economizer Upper Temperature Limit
Max OA%	0	100	%	=Maximum Percent Outside Air
Min OA%	0	100	%	=Minimum Percent Outside Air

Functional Relationships

There is a VBA function which relates the “Econo Type” Doe2.2 parameter to the input parameters “# of Economizer Sensors” and “Economizer Type.” All the other parameters are directly mapped to each other.

```
Public Function Econo_Type(EconomizerType As Long, NumEconomizerSensors As Long)

If EconomizerType = 1 Then
  If NumEconomizerSensors = 0 Then
     Econo_Type = 0
  ElseIf NumEconomizerSensors = 1 Then
```

```
      Econo_Type = 1
   ElseIf NumEconomizerSensors = 2 Then
      Econo_Type = 3
   Else
      MsgBox "Invalid entry for the number of economizer sensors! A value of " & _
      NumEconomizerSensors & " was provided the valid values are 0,1,2"
      Econo_Type = -1
   End If
ElseIf EconomizerType = 2 Then
   If NumEconomizerSensors = 0 Then
     Econo_Type = 0
   ElseIf NumEconomizerSensors = 1 Then
     Econo_Type = 2
   ElseIf NumEconomizerSensors = 2 Then
     Econo_Type = 4
   Else
      MsgBox "Invalid entry for the number of economizer sensors! A value of " & _
      NumEconomizerSensors & " was provided the valid values are 0,1,2"
      Econo_Type = -1
   End If
Else
      MsgBox "Invalid entry for the economizer type! A value of " & _
      EconomizerType & " was provided the valid values is 1 or 2"
      Econo_Type = -1
End If
End Function
```

How to implement the ECM in a new building:

1. In a building input file, for every SYSTEM command which contains an air side economizer insert the following expressions:

```
ECONO-LIMIT-T    = {#pa("Econo UpLimit T")}
MAX-OA-FRACTION  = {#pa("Max OA%")}
MIN-OUTSIDE-AIR  = {#pa("Min OA%")}
OA-CONTROL       =
{if(#pa("Econo Type")=0) then
  #si("FIXED","SYSTEM","OA-CONTROL")
 else if(#pa("Econo Type")=1) then
  #si("OA-TEMP","SYSTEM","OA-CONTROL")
 else if(#pa("Econo Type")=2) then
  #si("OA-ENTHALPY","SYSTEM","OA-CONTROL")
 else if(#pa("Econo Type")=3) then
```

```
 #si("DUAL-TEMP","SYSTEM","OA-CONTROL")
else if(#pa("Econo Type")=4) then
 #si("DUAL-ENTHALPY","SYSTEM","OA-CONTROL")
else
 #si("FIXED","SYSTEM","OA-CONTROL")
 endif endif endif endif
endif
}
```

2. At the beginning of the file insert the appropriate parameters “Econo Type”, “Econo UpLimit T”, “Max OA%”, “Min OA%.”

Boiler Thermal Efficiency

ECM Description: This ECM increases energy efficiency through allowing an increase to the efficiency of the boilers in a building. It is a simple one-input ECM.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Boiler Thermal Efficiency	0.1	1	efficiency	Thermal efficiency of the boilers in a building. Typical values are 0.75-0.85 with the most efficient boilers reaching toward 0.90. This is the amount of heat output for every unit of heat input to the boiler

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Boiler Thermal Eff	0.1	1	efficiency	Thermal efficiency of the boilers in a building. Typical values are 0.75-0.85 with the most efficient boilers reaching toward 0.90. This is the amount of heat output for every unit of heat input to the boiler

Functional Relationships

`Boiler Thermal Efficiency` = `Boiler Thermal Eff`

How to implement the ECM in a new building:

1. For every BOILER in the building input the following

HEAT-INPUT-RATIO = {1/#pa("Boiler Thermal Eff")}

Caulk & Weather Strip Doors & Windows

ECM Description: This ECM seeks to improve building energy performance by targeted application of caulking sealant and weather stripping to reduce air infiltration associated with doors and windows that result in unwanted outdoor air entering the building (weather strip infiltration air changes per hour - ACH) causing additional HVAC system load during both heating and cooling seasons.

There are two user input parameters for this ECM. It is best to use one or the other but not to use both. The first “Weather Strip Infiltration Air Changes Per Hour” directly alters the air changes per hour (ACH) due to leakage in the buildings. It can be used if a good estimate of the ACH is known. In cases where less is known about the ACH, the second parameter, “Weather Strip Infiltration Percent Reduction,” is useful. It can be used to reduce the existing ACH by a percentage. This is also the parameter of choice if many buildings are being changed simultaneously to produce a site-wide result since it looks at reduction in ACH from a relative perspective. It is advised to never change both parameters because the resulting ACH is harder to understand. The relationship in the building input files is:

Leakage ACH = Air Changes Per Hour * (1 – “Infiltration Percent Reduction”)

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Weather Strip Infiltration Air Changes Per Hour	0	20	Air Changes Per Hour	Air changes per hour that occur due to leaks in the building envelope. This parameter is usually left unchanged. Use “Weather Strip Infiltration Percent Reduction” instead. Avoid changing both parameters to minimize confusion.
Weather Strip Infiltration Percent Reduction	0	100	%	Percent reduction in air changes per hour due to applying caulking and weather-stripping.

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation

Name	Low Value	High Value	Units	Explanation
WStrip Infil ACH	0	20	Air Changes Per Hour	=”Weather Strip Infiltration Air Changes Per Hour”
WStrip Infil % Reduct	0	100	%	=”Weather Strip Infiltration Percent Reduction”

Functional Relationships

There are no significant functional relationships. The IX Input Parameters are equal to their corresponding eQUEST parameters. Within the building input file a relationship exists:

{#pa("WStrip Infil ACH") * local("VOLUME") / local("AREA") / 60
* (1 - #pa("WStrip Infil % Reduct"))}

How to implement the ECM in a new building:

1. For every perimeter SPACE command insert the following expression below. The desired result is in cubic feet per minute divided by the floor area

INF-FLOW/AREA =
{#pa("WStrip Infil ACH") * local("VOLUME") / local("AREA") / 60
* (1 - #pa("WStrip Infil % Reduct"))}

2. Make sure that the keyword INF-METHOD = AIR-CHANGE for all of the spaces altered in step 1.
3. Insert new parameters “WStrip Infil ACH” and “WStrip Infil % Reduct” at the beginning of the building input file with appropriate values

Chilled Water Temperature Reset

ECM Description: This ECM improves building energy performance through a control strategy that increases chilled water temperature relative to the cooling requirements of the building. This reduces energy consumption associated with maintaining a constant chilled water temperature when chilled water demand decreases.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Chilled Water Design	32	100	Fahrenheit	The minimum temperature which the chilled water loop is designed to

Name	Low Value	High Value	Units	Explanation
Minimum Temperature				deliver under maximum load conditions (i.e. summer)
Chilled Water Reset Maximum Temperature	32	100	Fahrenheit	Upper limit on the supply temperature when load-reset is active (i.e. “Include Chilled Water Reset?”=1) This is the maximum temperature of the chilled water when cooling loads are small.
Include Chilled Water Reset?	0	1	none	1 = Yes 0 = No

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
CHW Design T Min	32	100	Fahrenheit	Chilled Water Design Minimum Temperature
CHW Reset T Max	32	100	Fahrenheit	Chilled Water Reset Maximum Temperature
CHW Reset?	0	1	None	1 = Yes, 0 = No

Functional Relationships

One to one relationships exist for all of the parameters.

How to implement the ECM in a new building:

1. In each CIRCULATION-LOOP of TYPE = CHW insert the following keyword-key value expressions

```
DESIGN-COOL-T    = {#pa("CHW Design T Min")}

COOL-SETPT-CTRL  =

{If(#pa("CHW Reset?") = 0) then

#SI("FIXED","CIRCULATION-LOOP","COOL-SETPT-CTRL") else

#SI("LOAD-RESET","CIRCULATION-LOOP","COOL-SETPT-CTRL") endif}
```

MAX-RESET-T = {#pa("CHW Reset T Max")}

MIN-RESET-T = {#pa("CHW Design T Min")}

Hot Water Temperature Reset

ECM Description: This ECM improves building energy performance through a control strategy that decreases hot water temperature relative to the heating requirements of the building. This reduces energy consumption associated with maintaining a constant hot water temperature when hot water demand decreases.

IX Input Parameters:

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Hot Water Design Maximum Temperature	100	210	Fahrenheit	Maximum temperature which the hot water loop is designed to deliver under maximum load (i.e. winter)
Hot Water Reset Minimum Temperature	100	210	Fahrenheit	Lower limit on the supply temperature when load-reset is active (i.e. "Include Hot Water Reset?"=1) This is the minimum temperature of the hot water when heating loads are small
Include Hot Water Reset?	0	1	None	1 = Yes, 0 = No

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
HW Design T Max	100	210	Fahrenheit	Hot Water Design Maximum Temperature
HW Reset T Min	100	210	Fahrenheit	Hot Water Minimum Temperature
HW Reset?	0	1		1 = Yes, 0 = No

Functional Relationships

One to one relationships exist for all of the parameters.

How to implement the ECM in a new building:

1. In each CIRCULATION-LOOP of TYPE=HW insert the following keyword-key value expressions

```
DESIGN-HEAT-T    = {#pa("HW Design T Max")}

HEAT-SETPT-CTRL  =

{If(#pa("HW Reset?") = 0) then

#SI("FIXED","CIRCULATION-LOOP","HEAT-SETPT-CTRL") else

#SI("LOAD-RESET","CIRCULATION-LOOP","HEAT-SETPT-CTRL") endif}

MAX-RESET-T      = {#pa("HW Design T Max")}

MIN-RESET-T      = {#pa("HW Reset T Min")}
```

Cool Roof

ECM Description: Apply a highly reflective roof onto a building to reduce heat loads during summer. The degree of reflectivity is expressed through the solar reflectance index (SRI). The result usually saves electrical cooling energy savings but leads to energy losses in the winter time. The heat loads in the building can change whether a cool roof is best. A large heat-load inside a building makes it more desirable to apply a cool roof since the building has a larger amount of time for which cooling is needed.

http://energy.lbl.gov/coolroof/ref_01.htm

"The Solar Reflectance Index (SRI.) is a measure of the roof's ability to reject solar heat, as shown by a small temperature rise. It is defined so that a standard black (reflectance 0.05, emittance 0.90) is 0 and a standard white (reflectance 0.80, emittance 0.90) is 100. For example, the standard black has a temperature rise of 90 deg. F (50 deg. C) in full sun, and the standard white has a temperature rise of 14.6 deg. F (8.1 deg. C). Once the maximum temperature rise of a given material has been computed, the SRI can be computed by interpolating between the values for white and black."

IX Input Parameters:

Name	Low Value	High Value	Units
New Cool Roof SRI	0	120	None
Percent New Cool Roof	0	100	%

Parameters In eQUEST/doe2.2

Name	Low Value	High Value	Units
Roof Absorbtance	0.05	0.95	Fraction

Functional Relationships

A simple mixture law is used to relate the SRI, Percent New Cool Roof, and Roof Absorbtance

`Roof Absorbtance` = (0.95 - 0.0075*$`New Cool Roof SRI`)*(1 - `Percent New Cool Roof`) + (0.95 - 0.0075*`New Cool Roof SRI`)*`Percent New Cool Roof`

Warning! If only the "New Cool Roof SRI" is entered without changing the percent new cool roof (whose default is always 0) then the change to roof absorbtance will be 0.

How to implement the ECM in a new building:

1. Insert the parameter "Roof Absorbtance" at the beginning of the file and give it a value which reflects the current state of the roof of the building. If more than one roofing surface exists, then use a weighted average.
2. For every CONSTRUCTION command which is used by a roof surface, insert the following

```
ABSORPTANCE     = {#pa("Roof Absorbtance")}
```

CUB Chilled Water Temperature Reset

ECM Description: This ECM seeks to improve building energy performance by increasing the chilled water temperature relative to the cooling requirements of the building. As cooling load decreases, the chilled water temperature can be increased.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Apply Chilled Water Reset?	0	1	0 = No, 1 = Yes	Switch to apply the chilled water reset.

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
CW Reset?	0	1	0 = No, 1 = Yes	Switch to apply the chilled water reset.

Functional Relationships

`CW Reset?` = `Apply Chilled Water Reset?`

How to implement the ECM in a new building:

1. Create parameter "CW Reset?"
2. For a CUB building in every command, CIRCULATION-LOOP, of TYPE, "CW," for the keyword, COOL-SETPT-CTRL, insert the following:

   COOL-SETPT-CTRL =
   {If(#pa("CW Reset?") = 0) then
   #SI("FIXED","CIRCULATION-LOOP","COOL-SETPT-CTRL") else
   #SI("LOAD-RESET","CIRCULATION-LOOP","COOL-SETPT-CTRL") endif}

   You may also want to adjust values in the DESIGN-COOL-T and MIN-RESET-T keywords but for now they are not included in the ECM

CUB Chiller Efficiency

ECM Description: This ECM saves energy by introducing an increase in the efficiency of chillers in a CUB. This represents replacing old chillers with newer higher efficiency chillers. Up to 7 chillers were included in the ECM to allow for multiple chiller loops. If a building does not have 7 chillers, then extra parameters are present but do not affect anything. They are assigned a value of -1 to indicate that there is no active chiller being viewed.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Chiller 1 Efficiency	0	5	kW/ton	Chiller efficiency for the first chiller
Chiller 2 Efficiency	0	5	kW/ton	Chiller efficiency for the second chiller
Chiller 3 Efficiency	0	5	kW/ton	Chiller efficiency for the third chiller
Chiller 4 Efficiency	0	5	kW/ton	Chiller efficiency for the fourth chiller
Chiller 5 Efficiency	0	5	kW/ton	Chiller efficiency for the fifth chiller
Chiller 6 Efficiency	0	5	kW/ton	Chiller efficiency for the sixth chiller
Chiller 7 Efficiency	0	5	kW/ton	Chiller efficiency for the seventh chiller

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
CUB Chiller 1 kW/ton	0	5	kW/ton	Chiller efficiency for the first chiller
CUB Chiller 2 kW/ton	0	5	kW/ton	Chiller efficiency for the second chiller
CUB Chiller 3 kW/ton	0	5	kW/ton	Chiller efficiency for the third chiller
CUB Chiller 4 kW/ton	0	5	kW/ton	Chiller efficiency for the fourth chiller
CUB Chiller 5 kW/ton	0	5	kW/ton	Chiller efficiency for the fifth chiller
CUB Chiller 6 kW/ton	0	5	kW/ton	Chiller efficiency for the sixth chiller
CUB Chiller 7 kW/ton	0	5	kW/ton	Chiller efficiency for the seventh chiller

Functional Relationships

Direct assignment of variables

`CUB Chiller **X** kW/ton` = `Chiller **X** Efficiency` **X** = 1,2,..7

How to implement the ECM in a new building:

1. Create all of the eQUEST parameters in the table above
2. For each command, CHILLER, in the keyword, ELEC-INPUT-RATIO, input the following: ELEC-INPUT-RATIO = {#pa("CUB Chiller **X** kW/ton") * 3413 / 12000} where **X** = 1,2,..,Number of chillers

CUB Cooling Tower Fan Control

ECM Description: This ECM gives the opportunity to see energy savings between having 2-speed fans in a CUB loop cooling tower versus variable speed fans. A single binary switch is the only input.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Variable Speed Drive Cooling Tower Fan Type	0	1	none	1 = Variable speed drive fan, 0 = 2 speed fan

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
CUB VSD Clg Twr Fans?	0	1	none	1 = Variable speed drive fan, 0 = 2 speed fan

Functional Relationships

` CUB VSD Clg Twr Fans?` = ` Variable Speed Drive Cooling Tower Fan Type `

How to implement the ECM in a new building:

1. Insert a new parameter "CUB VSD Clg Twr Fans?" with a value of 0 or 1
2. In every cooling tower command, HEAT-REJECTION (TYPE = many different possibilities including FLUID-COOLER, OPEN-TWR) insert the following for the CAPACITY-CTRL keyword:

```
  CAPACITY-CTRL =
{If(#pa("CUB VSD Clg Twr Fans?") = 1) then
  #SI("VARIABLE-SPEED-FAN","HEAT-REJECTION","CAPACITY-CTRL")
 else
  #SI("TWO-SPEED-FAN","HEAT-REJECTION","CAPACITY-CTRL")
 endif}
```

3. In the same set of commands as for step 1, insert the following for the CELL-CTRL keyword:

```
CELL-CTRL   =
{If(#pa("CUB VSD Clg Twr Fans?") = 1) then
  #SI("MAX-CELLS","HEAT-REJECTION","CELL-CTRL")
 else
  #SI("MIN-CELLS","HEAT-REJECTION","CELL-CTRL")
 endif}
```

CUB Flat Plate Economizer Use

ECM Description: This ECM controls when a flat plate economizer is used. The "Flat Plate Heat Exchanger Maximum Operational Outside Air Temperature" is a threshold that dictates whether a flat plate economizer/heat exchanger is used in place of another chiller. The capacity of the plate and frame economizer can be increased and decreased using "Flat Plate Economizer Flow Capacity."

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Flat Plate Economizer	0	10000	Gallons per	Flow capacity of the flat plate heat

Flow Capacity			minute	exchanger being used.
Flat Plate Heat Exchanger Maximum Operational Outside Air Temperature	0	100	Fahrenheit	When the outdoor air temperature is above this value, a flat plate heat exchanger is no longer used

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
WSE Capacity (gpm)	0	10000	Gallons per minute	
WSE Max OA Temperature	0	100	Fahrenheit	

Functional Relationships

There are one to one relationships for all parameters.

How to implement the ECM in a new building:

1. Create new parameters "WSE Capacity (gpm)" and "WSE Max OA Temperature" and assign them appropriate values.
2. If needed create a new LOAD-MANAGEMENT command and make it of type "OA-TEMP."
3. In LOAD-MANAGEMENT commands insert the following for TEMPS-THRU-1 keyword:

```
TEMPS-THRU-1      = {#pa("WSE Max OA Temperature")}
```

4. If no plate and frame heat exchanger exists in the model create one through a CHILLER command.
5. Create a new EQUIP-CTRL command which includes the plate and frame heat exchanger
6. In CHILLER commands which are designed to be plate and frame heat exchangers (the user must determine this), place the following relationship in the CAPACITY keyword

```
CAPACITY          = {MAX(#pa("WSE Capacity (gpm)"),1)*F }
```

Where **F** is a factor which scales the capacity of the flat plate heat exchanger. The user must choose F carefully.

Daylighting Sensors for Top and Side Lighting

ECM Description: This ECM seeks to save energy by the installation of sensors that turn lights off when natural lighting is sufficient. Daylighting is one of many concepts being implemented to make lighting smarter: https://en.wikipedia.org/wiki/Smart_lighting

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Daylighting Control System	0	2	None	0 = no daylighting control, 1 = dimming (continuous), 2 = stepped
Minimum Dimming Ratio	0	100	%	Minimum level lighting will reduce when dimmed
Number Of Daylighting Steps	0	10	Integer	Number of steps which are used in dimming light if "Daylighting Control System" = 2 Otherwise this input is not needed
Percent Retrofitted With Daylighting	0	100	%	Percentage of lighting in the building which has been retrofitted with daylighting sensors

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
# DL Steps	0	10	Integer	0 = No daylighting, 1 = Continuous, 2,3,4 ..10 number of daylighting steps for a discreet system
%New DL Ctrls	0	100	%	Percentage of lighting in the building which has been retrofitted with daylighting sensors
DL Min Pwr	0	100	%	Minimum level lighting will reduce when dimmed

Functional Relationships

\`%New DL Ctrls\` = \`Percent Retrofitted With Daylighting\`
\`DL Min Pwr\` = \`Minimum Dimming Ratio\`

A VBA function is used to produce the output for "# DL Steps"

```
`# DL Steps` = DaylightingSteps(`Daylighting Control System`,`Number Of Daylighting Steps`)
```

```
Public Function DaylightingSteps(DLControlSystem As Long, NumberOfSteps As Long) As Double
Dim L As Long

If DLControlSystem <= 1 Then
  L = DLControlSystem
Else
  L = NumberOfSteps
End If

DaylightingSteps = L

End Function
```

<u>How to implement the ECM in a new building:</u>

1. Create the required eQUEST parameters for this ECM.
2. In every space or in the SET-DEFAULT FOR SPACE command insert the following for the LIGHT-CTRL-TYPE-1 keyword:

```
 LIGHT-CTRL-TYPE1 =
{if(#pa("# DL Steps")=1) then
  #si("CONTINUOUS","SPACE","LIGHT-CTRL-TYPE1")
 else if(#pa("# DL Steps")=0) then
  unused
 else if(#pa("# DL Steps")>1) then
  #si("STEPPED","SPACE","LIGHT-CTRL-TYPE1")
 else unused
 endif endif endif}
```

3. Same command context as step 2 for the DAYLIGHTING keyword:

```
DAYLIGHTING     =
{if(#pa("# DL Steps")=0) then
  #si("NO","SPACE","DAYLIGHTING")
 else
  #si("YES","SPACE","DAYLIGHTING")
 endif}
```

4. Same command context as step 2 for the LIGHT-CTRL-STEPS keyword:

```
LIGHT-CTRL-STEPS =
{if(#pa("# DL Steps")=1) then unused
 else if(#pa("# DL Steps")=0) then unused
 else if(#pa("# DL Steps")>1) then
  #pa("# DL Steps")-1
 else unused
 endif endif endif}
```

5. For the same command context as step 2 for the MIN-POWER-FRAC keyword:

```
MIN-POWER-FRAC   = {#pa("DL Min Pwr")}
```

6. For each space in the building for the ZONE-FRACTION1 keyword:

```
ZONE-FRACTION1    = {#pa("%New DL Ctrls") * F}
```

Where F is a fraction < 1 and > 0 which must be determined by the user based on how much lighting is available for daylight sensors in the space being considered (i.e. a room with no windows F = 0).

Duct Static Pressure Reset (repair & commission)

ECM Description: This ECM seeks to improve building energy performance by enabling or implementing a duct static pressure reset control strategy to lower duct static pressure in variable-air-volume systems based on optimizing supply fan speed and air flow rate, thereby reducing supply fan-power requirements as well as reducing waste heat and energy required for cooling.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Is VAV Static Press Reset In Use?	0	1	None	1 = Yes, 0 = No
VAV Box Average Minimum Flow Ratio	0	1	Ratio	Minimum air flow through the supply fan (not through the system), expressed as a fraction of the design flow rate. When a variable air volume system requires less air flow, a fan bypass is utilized.

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
VAV Box Avg Min Flow Ratio	0	1	Ratio	Minimum air flow through the supply fan (not through the system), expressed as a fraction of the design flow rate. When a variable air volume system requires less air flow, a fan bypass is utilized.
VAV Static Press Reset in Use	0	1	None	1 = Yes, 0 = No

Functional Relationships

One to one relationships

How to implement the ECM in a new building:

1. Create the required eQUEST parameters for this ECM
2. Insert the following in the ZONE default command or in every ZONE in the building:

```
MIN-FLOW-RATIO   = {#pa("VAV Box Avg Min Flow Ratio")}
```

3. Create two CURVE-FIT commands one named "Good SP Reset VSD Fan" and another "`Min SP Reset VSD Fan.`" "Good" signifies a fan which has a static pressure reset functioning. The user will have to figure out what coefficients to input but the typical values used are provided below:

```
"Min SP Reset VSD Fan" = CURVE-FIT
  TYPE           = CUBIC
  INPUT-TYPE      = COEFFICIENTS
  OUTPUT-MIN      = 0
  OUTPUT-MAX      = 1
  COEFFICIENTS    = ( 0.0704289, 0.38533, -0.460864, 1.0092 )
  ..
"Good SP Reset VSD Fan" = CURVE-FIT
  TYPE           = CUBIC
  INPUT-TYPE      = COEFFICIENTS
  OUTPUT-MIN      = 0
  OUTPUT-MAX      = 1
```

```
COEFFICIENTS     = ( 0.0407599, 0.088045, -0.0729261, 0.94374 )
..
```

4. For SYSTEMS of TYPE VAVS (you can use the SET-DEFAULT command) insert the following:

```
FAN-EIR-FPLR     =
{If(#pa("VAV Static Press Reset in Use") = 0) then
  #SI("Min SP Reset VSD Fan","SYSTEM","FAN-EIR-FPLR")
 else
  #SI("Good SP Reset VSD Fan","SYSTEM","FAN-EIR-FPLR")
 Endif
}

RETURN-EIR-FPLR  =
{If(#pa("VAV Static Press Reset in Use") = 0) then
  #SI("Min SP Reset VSD Fan","SYSTEM","FAN-EIR-FPLR")
 else
  #SI("Good SP Reset VSD Fan","SYSTEM","FAN-EIR-FPLR")
 Endif
}
```

Exterior Insulated Finish System (EIFS)

ECM Description: Add insulation to exterior walls of a building. This is actual insulation besides other layers in the exterior wall construction and does not represent the total R-Value for walls.

https://en.wikipedia.org/wiki/Exterior_insulation_finishing_system

https://en.wikipedia.org/wiki/R-value_%28insulation%29

IX Input Parameters:

Name	Low Value	High Value	Units
New EFIS R-Value	0	100	F-ft^2-hr/Btu
New EFIS Percent of Wall to Receive New Insulation	0	100	%

Parameters In eQUEST/Doe2.2

Name	Low Value	High Value	Units
EIFS Insul R-Value	0	100	F-ft^2-hr/Btu

Functional Relationships

A simple mixture law is used to relate calculate "New EFIS R-Value." "$" indicates that the original baseline value is retained for later years.

`EFIS Insul R-Value` = $`New EFIS R-Value`*(1 - `New EFIS Percent of Wall to Receive New Insulation`) + `New EFIS R-Value`*`New EFIS Percent of Wall to Receive New Insulation`

Warning! If only the "New EFIS R-Value" is entered without changing the "New EFIS Percent of Wall to Receive New Insulation" (whose default is always 0) then the change to "EIFS Insul R-Value" will be 0.

How to implement the ECM in a new building:

1. Insert the parameter "EIFS Insul R-Value" and assign it an appropriate value of 0 unless an EIFS has already been applied.
2. Create a new material called "EIFS R-Value Mat"
   ```
   "EIFS R-Value Mat" = MATERIAL
     TYPE          = RESISTANCE
     RESISTANCE      = {#pa("EIFS Insul R-Value") + 0.1}
     ..
   ```
3. For every exterior wall LAYERS command, insert the new material as the first entry in the layers

High Efficiency Lighting Replacements

ECM Description: This ECM seeks to improve building energy performance by using higher efficiency lighting than currently exists in a building to reduce building electricity consumption. Two models exist in this single ECM. The first combines a mixture of compact fluorescent, LED, T5, T8, and the original lighting type to reduce the average lighting power density. The second establishes a new lighting power density based on the ASHRAE 90.1 lighting power density standard for 2001, 2004, 2007, 2010, or 2013.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
% CFL	0	100	%	Percent of new lighting that will be compact fluorescent lighting. All of the % parameters are added up and normalized to 100% if they do not add up to exactly 100%

% LED	0	100	%	Percent of new lighting that will be LED lighting
% Original	0	100	%	Percent of lighting that remains the original lighting type. This could be a mixture of lighting technologies lumped into a single group
% T5	0	100	%	Percent of new lighting that will be T5 lighting
% T8	0	100	%	Percent of new lighting that will be T8 lighting
CFL LPD	0	10	W/m2	Lighting power density of compact fluorescent lighting
LED LPD	0	10	W/m2	Lighting power density of LED lighting
LPD Vintage	0	5	None	This parameter controls which of two models for this ECM is used whether the % lighting and lighting power density parameters 0 = By Percent Mixtures 1,2,3,4,5 = ASHRAE 90.1, 2001,2004,2007,2010,2013
Original LPD	0	10	W/m2	Original (current) lighting power density in the building
T5 LPD	0	10	W/m2	T5 fixtures lighting power density
T8 LPD	0	10	W/m2	T8 fixtures lighting power density

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Avg LPD	0	10	W/m2	The average lighting power density based on all of the user input parameters above. This parameter is applied throughout the building.

Functional Relationships

A VBA function relates "Avg LPD" to the user input parameters

CalculateAvgLPD(`LPD Vintage`, `Original LPD`, `CFL LPD`, `LED LPD`, `T8 LPD`, `T5 LPD`, `% Original`, `% CFL`, `% LED`, `% T8`, `% T5`,FunctionParameters,FunctionParameters)

```
Public Function CalculateAvgLPD(LPDVintage As Double, _
                OriginalLPD As Double, _
                CFLLPD As Double, _
                LEDLPD As Double, _
                T8LPD As Double, _
                T5LPD As Double, _
                PercOrg As Double, _
                PercCFL As Double, _
                PercLED As Double, _
                PercT8 As Double, _
                PercT5 As Double, _
                BuildingID As String, _
                BuildingFileID As Long) As Double
LPDVintage = Int(LPDVintage)

If LPDVintage = 0 Then ' Calculate based on user input.
    Dim SumPerc As Double

    SumPerc = PercOrg + PercCFL + PercLED + PercT8 + PercT5
    'Normalize percentages to add to one
    PercOrg = PercOrg / SumPerc
    PercCFL = PercCFL / SumPerc
    PercLED = PercLED / SumPerc
    PercT8 = PercT8 / SumPerc
    PercT5 = PercT5 / SumPerc
    'First normalize the entire
    CalculateAvgLPD = OriginalLPD * PercOrg + _
               CFLLPD * PercCFL + _
               LEDLPD * PercLED + _
               T8LPD * PercT8 + _
               T5LPD * PercT5
Else ' Figure out the lighting based on ashrae standards
     Dim Db As clsDatabase_shared
     Set Db = New clsDatabase_shared
```

```
    Dim LPDs As Variant
    mdlGlobalConstants.InitializeGlobalConstants

    If Application.Name = "Microsoft Excel" Then
      Db.DbName = IXdbName
    Else
      Db.DbName = CurrentProject.Path & "\" & CurrentProject.Name
    End If

    LPDs = Db.AccessQueryInDatabase("qryASHRAE_LPD_ByBuildingID", 0, BuildingFileID)
    CalculateAvgLPD = LPDs(LPDVintage)
End If

End Function
```

How to implement the ECM in a new building:

1. Insert a new parameter command and name the parameter “Avg LPD.”
2. In every SPACE command or in the SET-DEFAULT SPACE command insert the following:

   ```
   LIGHTING-W/AREA  = ( {#pa("Avg LPD")*#pa("LPD Mult")} )
   ```

   Here “LPD Mult” is a parameter that is used by the reduce illumination levels ECM but can be omitted if the reduce illumination levels ECM is not going to be included

Hot Water Pump Flow Control

ECM Description: This ECM seeks to reduce energy by allowing the user to specify the use of variable speed pumps for CUB primary and secondary hot water loops. This ECM needs updating because the IX user input parameters do not clearly specify in their names that the first is for a primary hot water loop and the second is for a secondary hot water loop. Otherwise both parameters accomplish the same task of either selecting a variable speed pump or a single speed pump.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Use Hot Water Loop Constant Volume Flow	0	1	none	For a CUB building’s primary hot water loop specify whether the pump is constant flow or variable flow. 1 = Constant Volume Flow, Anything else = Variable Flow

Use Hot Water Loop Variable Speed Drive Pump	0	1	none	For a CUB building's secondary hot water loop specify whether the pump is variable speed or constant flow. 1 = variable speed, Anything else = Single speed

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
CUB Pri HW Loop CV Flow?	0	1	none	1 = Constant Volume Flow, Anything else = Variable Volume Flow
CUB Sec HW Loop VSD Pump?	0	1	none	1 = variable speed, Anything else = Single speed

Functional Relationships

One to one relationships

How to implement the ECM in a new building:

1. Insert the parameters in the eQUEST parameter table above
2. For CUB buildings for each BOILER command, find the loop's PUMP command and then insert the following:

```
  CAP-CTRL       =
{If(#pa("CUB Pri HW Loop CV Flow?") = 1) then
 #SI("ONE-SPEED-PUMP","PUMP","CAP-CTRL")
 else
 #SI("VAR-SPEED-PUMP","PUMP","CAP-CTRL")
 endif}
```

3. Next for each BOILER insert the following

```
  HW-FLOW-CTRL    =
{If(#pa("CUB Pri HW Loop CV Flow?") = 1) then
 #SI("CONSTANT-FLOW","BOILER","HW-FLOW-CTRL")
 else
 #SI("VARIABLE-FLOW","BOILER","HW-FLOW-CTRL")
 endif}
```

4. Next find the secondary hot water loop and its corresponding PUMP command. Insert the following:

```
CAP-CTRL      =
{If(#pa("CUB Sec HW Loop VSD Pump?") = 1) then
  #SI("VAR-SPEED-PUMP","PUMP","CAP-CTRL")
 else
  #SI("ONE-SPEED-PUMP","PUMP","CAP-CTRL")
 endif}
```

Install Exterior Shading Overhangs

ECM Description: This ECM seeks to improve building energy performance by applying exterior shading devices (overhangs, louvers, awnings, etc.) that minimize direct sunlight on window surfaces during hot summer months when solar heat is undesirable, while allowing direct sunlight on window surfaces during colder winter months when solar heat is desirable.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Percent New Exterior Overhang East	0	100	%	Over hang width as a percentage of the local window's width for east facing windows which have overhangs added
Percent New Exterior Overhang North	0	100	%	Over hang width as a percentage of the local window's width for north facing windows which have overhangs added
Percent New Exterior Overhang South	0	100	%	Over hang width as a percentage of the local window's width for south facing windows which have overhangs added
Percent New Exterior Overhang West	0	100	%	Over hang width as a percentage of the local window's width for west facing windows which have overhangs added
New Exterior Overhang	0	20	feet	Length of overhangs measured

Depth	normal to the window surface

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
%New Ext Ovrhang E	0	100	%	Over hang width as a percentage of the local window's width for east facing windows which have overhangs added
%New Ext Ovrhang N	0	100	%	Over hang width as a percentage of the local window's width for north facing windows which have overhangs added
%New Ext Ovrhang S	0	100	%	Over hang width as a percentage of the local window's width for south facing windows which have overhangs added
%New Ext Ovrhang W	0	100	%	Over hang width as a percentage of the local window's width for west facing windows which have overhangs added
New Ext Ovrhang Depth	0	20	Feet	Length of overhangs measured normal to the window surface

Functional Relationships

One to one relationships

How to implement the ECM in a new building:

1. Insert the required eQUEST parameters seen in the table above.
2. In each WINDOW command or in the SET-DEFAULT FOR WINDOW command insert the following:

```
  OVERHANG-W     =
{if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
#g("BUILD-PARAMETERS","AZIMUTH"), 90) > 315.) then
  local("WIDTH") * #pa("%New Ext Ovrhang N")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
#g("BUILD-PARAMETERS","AZIMUTH"), 90) > 225.) then
```

```
  local("WIDTH") * #pa("%New Ext Ovrhang W")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
#g("BUILD-PARAMETERS","AZIMUTH"), 90) > 135.) then
  local("WIDTH") * #pa("%New Ext Ovrhang S")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
#g("BUILD-PARAMETERS","AZIMUTH"), 90) > 45.) then
  local("WIDTH") * #pa("%New Ext Ovrhang E")
else local("WIDTH") * #pa("%New Ext Ovrhang N")
endif endif endif endif}
  OVERHANG-D      = {#pa("New Ext Ovrhang Depth")}
```

Install Interior Shading

ECM Description: This ECM seeks to improve building energy performance by applying interior shading devices (blinds, shades, shutters, draperies, etc.) to windows that minimize the fraction of heat flow that enters the room (indoor solar attenuation coefficient – IAC), thereby reducing the internal heat load and corresponding energy use for air conditioning.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
New Interior Shading IAC	0	100	%	Indoor solar attenuation coefficient (IAC). 0% means that there is no shading, 100% means that no solar radiation gets through.
Percent Interior Shading East	0	100	%	Percent of east-facing windows which receive interior shading
Percent Interior Shading North	0	100	%	Percent of north-facing windows which receive interior shading
Percent Interior Shading South	0	100	%	Percent of south-facing windows which receive interior shading
Percent Interior Shading West	0	100	%	Percent of west-facing windows which receive interior shading
Interior Shading Probability of Use	0	1	Fraction	Probability interior shades will be used

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Int Shade Prob of Use	0	1	fraction	Probability that shade will be used
IAC N	0	1	fraction	Indoor solar attenuation coefficient (IAC) for northward facing windows
IAC S	0	1	fraction	Indoor solar attenuation coefficient (IAC) for southward facing windows
IAC W	0	1	fraction	Indoor solar attenuation coefficient (IAC) for westward facing windows
IAC E	0	1	fraction	Indoor solar attenuation coefficient (IAC) for eastward facing windows

Functional Relationships

Simple mixture laws for N, E, S, and W (remember “$” indicates that a value stays at the baseline year value). Example for west:

$`New Interior Shading IAC`*$`Percent Interior Shading West`*(1 - `Percent Interior Shading West`) + `New Interior Shading IAC` * `Percent Interior Shading West`

` Int Shade Prob of Use ` = `Interior Shading Probability of Use`

How to implement the ECM in a new building:

1. Insert the 5 parameters eQUEST parameter in the table above.
2. Insert the following in every WINDOW command or in the SET-DEFAULT FOR WINDOW command

```
  SHADING-SCHEDULE =
{if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
    #g("BUILD-PARAMETERS","AZIMUTH"), 90)  > 315.) then
  #si("Window N Int Shade Sch","WINDOW","SHADING-SCHEDULE")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
    #g("BUILD-PARAMETERS","AZIMUTH"), 90)  > 225.) then
  #si("Window W Int Shade Sch","WINDOW","SHADING-SCHEDULE")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
```

```
    #g("BUILD-PARAMETERS","AZIMUTH"), 90)  > 135.) then
  #si("Window S Int Shade Sch","WINDOW","SHADING-SCHEDULE")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
    #g("BUILD-PARAMETERS","AZIMUTH"), 90)  > 45.) then
  #si("Window E Int Shade Sch","WINDOW","SHADING-SCHEDULE")
else #si("Window N Int Shade Sch","WINDOW","SHADING-SCHEDULE")
endif endif endif endif}
  SUN-CTRL-PROB    = {#pa("Int Shade Prob of Use")}
```

3. Create the 4 schedules Window E Int Shade Sch, Window N Int Shade Sch, Window S Int Shade Sch, and Window W Int Shade Sch.

```
"Window N Int Shade Sch" = SCHEDULE-PD
  TYPE             = FRACTION
  MONTH            = ( 12 )
  DAY              = ( 31 )
  WEEK-SCHEDULES   = ( "Window N Int Shade WSch" )
  ..
"Window S Int Shade Sch" = SCHEDULE-PD
  TYPE             = FRACTION
  MONTH            = ( 12 )
  DAY              = ( 31 )
  WEEK-SCHEDULES   = ( "Window S Int Shade WSch" )
  ..
"Window E Int Shade Sch" = SCHEDULE-PD
  TYPE             = FRACTION
  MONTH            = ( 12 )
  DAY              = ( 31 )
  WEEK-SCHEDULES   = ( "Window E Int Shade WSch" )
  ..
"Window W Int Shade Sch" = SCHEDULE-PD
  TYPE             = FRACTION
  MONTH            = ( 12 )
  DAY              = ( 31 )
  WEEK-SCHEDULES   = ( "Window W Int Shade WSch" )
```

4. Create the WEEK-SCHEDULES for E, N, S, and W.

```
"Window N Int Shade WSch" = WEEK-SCHEDULE-PD
  TYPE             = FRACTION
  DAY-SCHEDULES    = ( "Window N Int Shade DSch" )
  ..
```

```
"Window S Int Shade WSch" = WEEK-SCHEDULE-PD
  TYPE            = FRACTION
  DAY-SCHEDULES   = ( "Window S Int Shade DSch" )
  ..
"Window E Int Shade WSch" = WEEK-SCHEDULE-PD
  TYPE            = FRACTION
  DAY-SCHEDULES   = ( "Window E Int Shade DSch" )
  ..
"Window W Int Shade WSch" = WEEK-SCHEDULE-PD
  TYPE            = FRACTION
  DAY-SCHEDULES   = ( "Window W Int Shade DSch" )
  ..
```

5. Create the day schedules for E, N, S, and W.

```
"Window N Int Shade DSch" = DAY-SCHEDULE-PD
  TYPE            = FRACTION
  VALUES          = ( {#pa("IAC N")} )
  ..
"Window S Int Shade DSch" = DAY-SCHEDULE-PD
  TYPE            = FRACTION
  VALUES          = ( {#pa("IAC S")} )
  ..
"Window E Int Shade DSch" = DAY-SCHEDULE-PD
  TYPE            = FRACTION
  VALUES          = ( {#pa("IAC E")} )
  ..
"Window W Int Shade DSch" = DAY-SCHEDULE-PD
  TYPE            = FRACTION
  VALUES          = ( {#pa("IAC W")} )
  ..
```

Install Revolving Doors or Vestibules

ECM Description: This ECM seeks to reduce energy by adding a structure that reduces the amount of outside air entering a building through the closing and opening of doors. A vestibule is a small room that is used as an entry way to a building. Revolving doors can also accomplish the same effect.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation

Percent Reduction Vestibule Infiltration	0	100	%	This is a percentage reduction in the number of air changes which occur due to addition of a vestibule or other item which reduces losses at entrances.
Vestibule Infiltration Air Changes Per Hour	0	2	Air Changes Per Hour	This is the number of air changes per hour which occur due to entrances into the building. Changes to both user parameters are unnecessary for this ECM.

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Vest Infil % Reduct	0	100	%	This is a percentage reduction in the number of air changes which occur due to addition of a vestibule or other item which reduces losses at entrances.
Vest Infil ACH	0	2	Air Changes Per Hour	This is the number of air changes per hour which occur due to entrances into the building. Changes to both user parameters are unnecessary for this ECM.

Functional Relationships

One to one relationships

How to implement the ECM in a new building:

1. Create the 2 eQUEST parameters above
2. For every SPACE command which contains a DOOR insert the following:

```
  INF-FLOW/AREA   =
{#pa("Vest Infil ACH") * local("VOLUME") / local("AREA") / 60
* (1 - #pa("Vest Infil % Reduct"))}
```

Insulate Roof

ECM Description: Change the roof insulation of a building. This is actual insulation besides other layers needed to construct the roof and does not represent the total roof R-value. R-values are a measure of how well movement of heat is resisted.

https://en.wikipedia.org/wiki/R-value_%28insulation%29

IX Input Parameters:

Name	Low Value	High Value	Units
New Roof Insulation Effective R-Value	0	100	F-ft^2-hr/Btu
Percent of Roof to Receive New Insulation	0	100	%

Parameters In eQUEST/doe2.2

Name	Low Value	High Value	Units
Roof Insul R-Value	0	100	F-ft^2-hr/Btu

Functional Relationships

A simple mixture law is used. "$" indicates that the original baseline value is retained for later years.

`Roof Insul R-Value` = $`New Roof Insulation Effective R-Value`*(1 - `Percent of Roof to Receive New Insulation`) + `New Roof Insulation Effective R-Value`*`Percent of Roof to Receive New Insulation`

Warning! If only the "New Roof Insulation Effective R-Value" is entered without changing the "Percent of Roof to Receive New Insulation (whose default is always 0) then the change to "Roof Insul R-Value" will be 0.

How to implement the ECM in a new building:

1. Insert the parameter "Roof Insul R-Value" and assign it an appropriate value.
2. Create a new material called "Roof R-Value Mat"
   ```
   "Roof R-Value Mat" = MATERIAL
     TYPE          = RESISTANCE
     RESISTANCE      = {#pa("Roof Insul R-Value") + 0.1}
     ..
   ```
3. For every roof LAYERS command, insert the new material one of the entries in the layers

Lab Exhaust

ECM Description: This ECM seeks to improve building energy performance by defining reduced laboratory space exhaust requirements (operating hours, air speed, air volume, etc.) in buildings, thereby reducing HVAC system (heating and cooling) energy consumption.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Lab Exhaust Control	0	1	none	0 = constant volume with no air flow control, 1 = control is by varying motor speed
Terminal Minimum Occupied Average Flow Fraction	0	1	fraction	The minimum amount of flow as a fraction of fully open flow which is allowed for lab exhaust during occupied periods
Terminal Minimum Unoccupied Average Flow Fraction	0	1	fraction	The minimum amount of flow as a fraction fully open flow which is allowed for lab exhaust during unoccupied periods

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Lab Exhaust Ctrl	0	1	none	0 = constant volume with no air flow control, 1 = control is by varying motor speed
Term Min Occ Avg Flow Frac	0	1	fraction	The minimum amount of flow as a fraction fully open flow which is allowed for lab exhaust during occupied periods
Term Min UnOcc Avg Flow Frac	0	1	fraction	The minimum amount of flow as a fraction fully open flow which is allowed for lab exhaust during unoccupied periods

Functional Relationships

One to one relationships.

How to implement the ECM in a new building:

1. Insert the 3 eQUEST parameters above.
2. In ZONE commands which contain Lab Exhaust equipment insert the following:

```
EXHAUST-FAN-CTRL =
{if(#pa("Lab Exhaust Ctrl") == 0) then
  #si("CONSTANT-VOLUME","ZONE","EXHAUST-FAN-CTRL")
 else if(#pa("Lab Exhaust Ctrl") == 1) then
  #si("SPEED","ZONE","EXHAUST-FAN-CTRL")
 else
  #si("CONSTANT-VOLUME","ZONE","EXHAUST-FAN-CTRL")
 endif endif}
EXHAUST-FAN-SCH  =
{if(local("EXHAUST-FAN-CTRL") <> #si("SPEED","ZONE","EXHAUST-FAN-CTRL")) then
  #si("Exhaust Sch","ZONE","EXHAUST-FAN-SCH")
 else
  #si("Term ECM Min Flow Sch","ZONE","EXHAUST-FAN-SCH")
 endif }
```

3. Create two day schedules which represent the occupied and unoccupied schedules for the building

```
"Term ECM Min Flow Occ DSch" = DAY-SCHEDULE-PD
  TYPE          = FRACTION
  VALUES        = ( {#pa("Term Min UnOcc Avg Flow Frac")}, &D, &D, &D, &D,
      &D, {#pa("Term Min Occ Avg Flow Frac")}, &D, &D, &D, &D, &D, &D, &D,
      &D, &D, &D, &D, {#pa("Term Min UnOcc Avg Flow Frac")} )
  ..
"Term ECM Min Flow UnOcc DSch" = DAY-SCHEDULE-PD
  TYPE          = FRACTION
  VALUES        = ( {#pa("Term Min UnOcc Avg Flow Frac")} )
```

4. Insert a weekly schedule for the two day schedules

```
"Term ECM Min Flow WSch" = WEEK-SCHEDULE-PD
  TYPE          = FRACTION
  DAY-SCHEDULES   = ( "Term ECM Min Flow Occ DSch", &D, &D, &D, &D,
      "Term ECM Min Flow UnOcc DSch" )
```

5. Insert a yearly schedule

```
"Term ECM Min Flow Sch" = SCHEDULE-PD
  TYPE          = FRACTION
  MONTH         = ( 12 )
  DAY           = ( 31 )
  WEEK-SCHEDULES  = ( "Term ECM Min Flow WSch" )
```

6. Be sure that the default “Exhaust Sch” also exists and create appropriate week and day schedules also as needed.

Limit Personal Space Heater Use

ECM Description: This ECM seeks to improve building energy performance by limiting the electricity consumption associated with personal space heaters through mandated space heater products, such as 400 Watt limit, occupancy sensor controlled power strip use, etc., as well as prohibiting personal space heater use, thereby reducing the “equipment power density” of a building.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Fraction of Zones With Space Heaters In Use	0	1	Fraction	Fraction of the zones in the building which have a space heater
Night Thermostat Heating Change	0	20	Fahrenheit	Increase in temperature due to space heater operations at night
Occupied Thermostat Heating Change	0	20	Fahrenheit	Increase in temperature due to space heater operations during workdays
Space Heater Typical Capacity	0	10000	Watts	Average space heater power rating for the entire building
Weekend Thermostat Heating Change	0	20	Fahrenheit	Increase in temperature due to space heater operations during the weekend

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Frac Ofcs w SpcHtrs	0	1	Fraction	Fraction of the zones in the building

Name	Low Value	High Value	Units	Explanation
				which have a space heater
SpcHtr Nite TStat+	0	20	Fahrenheit	Increase in temperature due to space heater operations at night
SpcHtr Occ TStat+	0	20	Fahrenheit	Increase in temperature due to space heater operations during workdays
SpcHtr Typical Capacity (W)	0	10000	Watts	Average space heater power rating for the entire building
SpcHtr WEH TStat+	0	20	Fahrenheit	Increase in temperature due to space heater operations during the weekend

Functional Relationships

One to one relationships

How to implement the ECM in a new building:

1. Create the 5 eQUEST parameters in the table above also create the parameters for the Thermostat Management and VAV Box Occupancy Sensors ECM's (these ECM's are intertwined and have to be created simultaneously).
2. Insert the following into every ZONE command or the SET-DEFAULT FOR ZONE command:

```
BASEBOARD-RATING =
{If(local("HEAT-TEMP-SCH") =
   #SI("SpHtr Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")) then
    #pa("SpcHtr Typical Capacity (W)") * (-3.413) else 0 endif}
  HEAT-TEMP-SCH    =
{ If(local("DESIGN-COOL-T")-INT(local("DESIGN-COOL-T"))=0) then

  If(local("DESIGN-HEAT-T")-INT(local("DESIGN-HEAT-T"))=0) then
   #SI("Occ Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
  else
    If(#pa("Frac Ofcs w SpcHtrs") >= INT((local("DESIGN-HEAT-T")-
      INT(local("DESIGN-HEAT-T")))*100)/100 ) then
    #SI("SpHtr Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
    else
    #SI("Occ Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
    endif
  endif
```

```
else

 If(#pa("Frac Offices Unoc") >= INT((local("DESIGN-COOL-T")-
   INT(local("DESIGN-COOL-T")))*100)/100 ) then

   If(local("DESIGN-HEAT-T")-INT(local("DESIGN-HEAT-T"))=0) then
    #SI("UnOcc Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
   else
     If(#pa("Frac Ofcs w SpcHtrs") >= INT((local("DESIGN-HEAT-T")-
       INT(local("DESIGN-HEAT-T")))*100)/100 ) then
      #SI("SpHtr Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
     else
       #SI("UnOcc Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
     endif
   endif

 else
   If(local("DESIGN-HEAT-T")-INT(local("DESIGN-HEAT-T"))=0) then
    #SI("Occ Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
   else
     If(#pa("Frac Ofcs w SpcHtrs") >= INT((local("DESIGN-HEAT-T")-
       INT(local("DESIGN-HEAT-T")))*100)/100 ) then
      #SI("SpHtr Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
     else
      #SI("Occ Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
     endif
   endif
 endif

endif }
```

3. Create schedules “SpHtr Heat T-Stat Sch.” If necessary also create “Occ Heat T-Stat Sch”

```
"SpHtr Heat T-Stat Sch" = SCHEDULE-PD
  TYPE           = TEMPERATURE
  MONTH          = ( 12 )
  DAY            = ( 31 )
  WEEK-SCHEDULES   = ( "SpHtr Heat T-Stat WSch" )
```

4. Create weekly schedule “SpHtr Heat T-Stat WSch”

```
"SpHtr Heat T-Stat WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "SpHtr Heat T-Stat WD DSch", &D, &D, &D, &D,
      "SpHtr Heat T-Stat WEH DSch" )
```

5. Create daily schedules for weekend days and weekdays:

```
"SpHtr Heat T-Stat WD DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES         = (
{#pa("Heating UnOcc T-Stat Setpoint")+#pa("SpcHtr Nite TStat+")}, &D, &D, &D,
      &D, &D,
      {#pa("Heating Occ T-Stat Setpoint")+#pa("SpcHtr Occ TStat+")}, &D,
      &D, &D, &D, &D, &D, &D, &D, &D, &D, &D,
      {#pa("Heating UnOcc T-Stat Setpoint")+#pa("SpcHtr Nite TStat+")} )
  ..
"SpHtr Heat T-Stat WEH DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES         = (
{#pa("Heating UnOcc T-Stat Setpoint")+#pa("SpcHtr WEH TStat+")} )
  ..
```

6. In every ZONE command or in the SET-DEFAULT FOR ZONE command insert the following:

```
  BASEBOARD-RATING =
{If(local("HEAT-TEMP-SCH") =
   #SI("SpHtr Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")) then
    #pa("SpcHtr Typical Capacity (W)") * (-3.413) else 0 endif}
```

New Window or Glass Properties

<u>ECM Description:</u> This ECM seeks to improve building energy performance by applying windows and/or glass with specific characteristics that increase insulation (U-value) and thereby reduce heat transfer through the window, minimize solar heat transmitted through the window and into the building interior (Solar Heat Gain Coefficient), and control the amount of visible light passing through the window (T-vis) for occupant comfort. Desired characteristics are generally orientation specific (North-South-East-West).

Properties that can be controlled are heat transfer coefficient (U-Value), Visible Transmittance (VT), and Solar Heat Gain Coefficient (SHGC). The intent is to capture with one set of inputs what may be thought of as three different ECMs: new windows, the addition of interior window films, and exterior solar screens to existing windows. These three ECMs, new windows, window film, or solar screens, are mutually exclusive and require change of U-value, SHGC, and VT. It is therefore necessary to only apply one of the ECMs at a time (i.e. there is no way to apply a new window and put a solar screen on the window). Like any other insulating portion of a building, windows have a tradeoff with respect to building heat loads that may make "better" windows lose rather than gain energy.

Solar Heat Gain Coefficient = "that fraction of incident solar radiation that actually enters a building through the entire window assembly as heat gain" http://www.commercialwindows.org/shgc.php

eQUEST and doe2.2 documentation contain a significant window properties library.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
New Window Percent East	0	100	%	The percentage of windows facing east that will have the new window type installed
New Window Percent North	0	100	%	The percentage of windows facing north that will have the new window type installed
New Window Percent South	0	100	%	The percentage of windows facing south that will have the new window type installed
New Window Percent West	0	100	%	The percentage of windows facing west that will have the new window type installed
Old Window SHGC East	0	1	none	Eastward facing current SHGC
Old Window SHGC North	0	1	none	Northward facing current SHGC
Old Window	0	1	none	Southward facing

SHGC South				current SHGC
Old Window SHGC West	0	1	none	Westward facing current SHGC
New Window SHGC	0	1	none	The new window's SHGC
Old Windows U-Value East	0.1	1.5	Btu/(hr*ft^2*F)	Eastward facing current U-Value
Old Windows U-Value North	0.1	1.5	Btu/(hr*ft^2*F)	Northward facing current U-Value
Old Windows U-Value South	0.1	1.5	Btu/(hr*ft^2*F)	Southward facing current U-Value
Old Windows U-Value West	0.1	1.5	Btu/(hr*ft^2*F)	Westward facing current U-Value
New Window U-Value	0.1	1.5	Btu/(hr*ft^2*F)	The new window's U-value
Old Windows VT East	0	1	none	Eastward facing current VT
Old Windows VT North	0	1	none	Northward facing current VT
Old Windows VT South	0	1	none	Southward facing current VT
Old Windows VT West	0	1	none	Westward facing current VT
New Window VT	0	1	none	The new window's VT

Parameters In eQUEST/doe2.2

Name	Low Value	High Value	Units
Window SHGC E	0	1	none
Window SHGC N	0	1	none

Window SHGC S	0	1	none
Window SHGC W	0	1	none
Window U-Value E	0.1	1.5	Btu/(hr*ft^2*F)
Window U-Value N	0.1	1.5	Btu/(hr*ft^2*F)
Window U-Value S	0.1	1.5	Btu/(hr*ft^2*F)
Window U-Value W	0.1	1.5	Btu/(hr*ft^2*F)
Window VT E	0	1	none
Window VT N	0	1	none
Window VT S	0	1	none
Window VT W	0	1	none

Functional Relationships

Simple mixture laws are used for North, East, South and West:

`Window SHGC E ` = `New Window SHGC`*`New Window Percent East` + `Old Window SHGC East`*(1 - `New Window Percent East`)

`Window U-Value E` = `New Window U-Value`*`New Window Percent East` + `Old Windows U-Value East`*(1 - `New Window Percent East`)

`Window VT E` = `New Window VT`*`New Window Percent East` + `Old Window VT East`*(1 - `New Window Percent East`)

Warning! You have to enter the New Window Percent East (North, South, West) with other New Window parameters in order to make a difference.

How to implement the ECM in a new building:

1. Insert all parameters from the eQUEST parameters table above.
2. This procedure assumes that you are NOT working with a CUB building input file which already has several buildings combined. You must start from individual buildings because the CUB file already has eliminated SET-DEFAULT commands.
3. At the beginning of the building input file, create all of the parameters in the table “Parameters in eQUEST/doe2.2” above (starting with Window SHGC E)
4. Create 4 GLASS-TYPE commands for North, South, East, and West with the parameter expressions as seen below

```
"North Glass Type" = GLASS-TYPE
  TYPE          = SHADING-COEF
  SHADING-COEF    = {#pa("Window SHGC N") / 0.87}
  GLASS-CONDUCT   = {1/(#pa("Window U-Value N") - 0.2)}
  VIS-TRANS       = {#pa("Window VT N")}
```

5. If one does not already exist, Insert a default command block for windows as follows. If one does already exist, then only change the GLASS-TYPE keyword by inserting the expressions for GLASS-TYPE below.

```
SET-DEFAULT FOR WINDOW
  GLASS-TYPE     =
{if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
    #g("BUILD-PARAMETERS","AZIMUTH"), 90)  > 315.) then
  #si("North Glass Type","WINDOW","GLASS-TYPE")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
    #g("BUILD-PARAMETERS","AZIMUTH"), 90)  > 225.) then
  #si("West Glass Type","WINDOW","GLASS-TYPE")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
    #g("BUILD-PARAMETERS","AZIMUTH"), 90)  > 135.) then
  #si("South Glass Type","WINDOW","GLASS-TYPE")
else if ( #CDN(#p("AZIMUTH")+#p2("AZIMUTH")+#p3("AZIMUTH")+
    #g("BUILD-PARAMETERS","AZIMUTH"), 90)  > 45.) then
  #si("East Glass Type","WINDOW","GLASS-TYPE")
else #si("North Glass Type","WINDOW","GLASS-TYPE")
endif endif endif endif}
  ..
```

6. Search every WINDOW command and eliminate any GLASS-TYPE keywords so that the default GLASS-TYPE command applies.

Night Cooling

ECM Description: This ECM seeks to improve building energy performance by using cool outside air to lower indoor air temperatures in the building during nighttime, unoccupied hours that then delay or minimize HVAC system cooling loads during the following day, thereby reducing cooling system energy consumption.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Enable Venting at Night	0	1	None	0 = no venting at night, 1 = venting at night
Night Venting Indoor Start Temperature	20	100	Fahrenheit	Minimum temperature set point for natural venting. This temperature is generally below the temperature in the cooling temperature schedule of the building
Night Venting Inflow to Outflow Change in Temperature	1	20	Fahrenheit	Minimum number of degrees that the outside dry-bulb temperature must be below the temperature of the system's control zone for the night ventilation fans to operate. The value is usually set equal to the temperature rise across the ventilation fans plus a couple of degrees to ensure that a reasonable cooling capacity is available when night venting is used.

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Enable Night Venting	0	1	None	0 = no venting at night, 1 = venting at night
Night Venting Indoor Start T	20	100	Fahrenheit	Minimum temperature set point for natural venting. This temperature is generally below the temperature in the cooling temperature schedule of the building
Night Venting In-Out DT	1	20	Fahrenheit	Minimum number of degrees that the outside dry-bulb temperature must be below the temperature of the system's control zone for the night ventilation fans to operate. The value is usually set equal to the temperature rise across the

ventilation fans plus a couple of degrees to ensure that a reasonable cooling capacity is available when night venting is used.

Functional Relationships

One-to-one relationships

How to implement the ECM in a new building:

1. Create the 3 eQUEST parameters in the table above.
2. Create three day schedules as follows

```
"Night Venting WD DSch" = DAY-SCHEDULE-PD
  TYPE          = ON/OFF
  VALUES        = ( {#pa("Enable Night Venting")}, &D, &D, &D, &D, &D, 0,
      &D, &D, &D, &D, &D, &D, &D, &D, &D, &D, &D,
      {#pa("Enable Night Venting")} )
  ..
"Night Venting Sat DSch" = DAY-SCHEDULE-PD
  TYPE          = ON/OFF
  VALUES        = ( 0 )
  ..
"Night Venting Sun DSch" = DAY-SCHEDULE-PD
  TYPE          = ON/OFF
  VALUES        = ( 0, &D, &D, &D, &D, &D, &D, &D, &D, &D, &D, &D, &D, &D,
      &D, &D, &D, &D, {#pa("Enable Night Venting")} )
```

3. Create two week schedules which uses the day schedules:

```
"Night Venting Enabled WSch" = WEEK-SCHEDULE-PD
  TYPE          = ON/OFF
  DAY-SCHEDULES   = ( "Night Venting WD DSch", &D, &D, &D, &D,
      "Night Venting Sat DSch", "Night Venting Sun DSch" )
  ..

"Night Venting Disabled WSch" = WEEK-SCHEDULE-PD
  TYPE          = ON/OFF
  DAY-SCHEDULES   = ( "Night Venting Sat DSch" )
  ..
```

4. Create a yearly schedule which uses the week schedules:

```
"Night Venting Sch" = SCHEDULE-PD
  TYPE           = ON/OFF
  MONTH          = ( 4, 9, 12 )
  DAY            = ( 30, 15, 31 )
  WEEK-SCHEDULES   = ( "Night Venting Disabled WSch",
      "Night Venting Enabled WSch", "Night Venting Disabled WSch" )
```

Adjust the "MONTH" and "DAY" intervals to appropriate values for the climate of the building. In this example, night venting is used from April 30th to September 15th.

5. Create another day schedule:

```
"Night Venting Tmin DSch" = DAY-SCHEDULE-PD
  TYPE           = TEMPERATURE
  VALUES         = ( {#pa("Night Venting Indoor Start T")} )
```

6. Create another week schedule:

```
"Night Venting Tmin WSch" = WEEK-SCHEDULE-PD
  TYPE           = TEMPERATURE
  DAY-SCHEDULES    = ( "Night Venting Tmin DSch" )
```

7. Create another yearly schedule:

```
"Night Venting Tmin Sch" = SCHEDULE-PD
  TYPE           = TEMPERATURE
  MONTH          = ( 12 )
  DAY            = ( 31 )
  WEEK-SCHEDULES   = ( "Night Venting Tmin WSch" )
```

8. In every air handling unit which will have night cooling enabled insert the following keywords:

```
  VENT-TEMP-SCH    = "Night Venting Tmin Sch"
  NIGHT-CYCLE-CTRL = CYCLE-ON-ANY
  NIGHT-VENT-CTRL  = SCHEDULED+DEMAND
  NIGHT-VENT-SCH   = "Night Venting Sch"
  NIGHT-VENT-DT    = {#pa("Night Venting In-Out DT")}
  NIGHT-VENT-RATIO = ( 0.3 )
```

Reduce Domestic Hot Water Recirculation Hours

ECM Description: This ECM seeks to reduce energy consumption by including a timer that turns off domestic hot water circulation during night hours.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Is a Domestic Hot Water Recirculation Pump Timer Used?	0	1	None	0 = no, 1 = yes

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
DHW Recirc Pump Timer?	0	1	None	0 = no, 1 = yes

Functional Relationships

One to one relationship

How to implement the ECM in a new building:

1. Create the parameter "DHW Recirc Pump Timer?"
2. Insert a day schedule:

```
“DHW Pump DSch" = DAY-SCHEDULE-PD
  TYPE          = ON/OFF
  VALUES         = (
{If(#pa("DHW Recirc Pump Timer?")=0) then
 1 else 0 endif}, &D, &D, &D, &D, &D, 1, &D, &D, &D, &D, &D, &D, &D, &D, &D,
      &D, &D,
      {If(#pa("DHW Recirc Pump Timer?")=0) then
 1 else 0 endif} )
  ..
```

3. Insert a week schedule

```
"DHW Pump WSch" = WEEK-SCHEDULE-PD
  TYPE          = ON/OFF
```

```
DAY-SCHEDULES    = ( "DHW Pump DSch" )
..
```

4. Insert a year schedule

```
"DHW Pump Sch" = SCHEDULE-PD
  TYPE           = ON/OFF
  MONTH          = ( 12 )
  DAY            = ( 31 )
  WEEK-SCHEDULES   = ( "DHW Pump WSch" )
```

5. In every hot water and domestic hot water loop insert the following KEYWORD:

```
PUMP-SCHEDULE    = "DHW Pump Sch"
```

Reduce Fan Operation Hours

ECM Description: This ECM seeks to improve building energy performance by establishing specific operating hours for building HVAC system fans (supply, return, exhaust, etc.) using a building automation system (BAS) that eliminates fan operation when not needed (non-occupied hours, for example), thereby reducing HVAC system fan power energy consumption.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Fan 24-7 Start Hour	0	24	hr	Hour of the day when fans start for zones that run 24-7. This is usually equal to 0 since the fans always run for 24-7 operations. This parameter gives the opportunity to change 24-7 operations to more limited operations. **Warning!** For any given building, there may or may not be zones that run in 24-7 mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.
Fan 24-7 Stop Hour	0	24	Hr	Hour of the day when fans stop for zones that run 24-7. This is usually equal to 24 since the fans always run for 24-7 operations. This parameter

				gives the opportunity to change 24-7 operations to more limited operations. **Warning!** For any given building, there may or may not be zones that run in 24-7 mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.
Fan Week Day Start Hour	0	24	Hr	Hour of the day at which fans are turned on during weekdays for all non 24-7 zones in a building. **Warning!** For any given building, there may or may not be zones that run in normal weekday mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.
Fan Week Day Stop Hour	0	24	Hr	Hour of the day at which fans are turned off during weekdays for all non 24-7 zones in a building. **Warning!** For any given building, there may or may not be zones that run in normal weekday mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.
Fan Week End Start Hour	0	24	Hr	Hour of the day at which fans are turned on during weekends for all non 24-7 zones in a building. **Warning!** For any given building, there may or may not be zones that run in normal weekend mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.

Fan Week End Stop Hour	0	24	Hr	Hour of the day at which fans are turned off during weekends for all non 24-7 zones in a building. **Warning!** For any given building, there may or may not be zones that run in normal weekend mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.

<u>Parameters In eQUEST/doe2.2 (read only):</u>

Name	Low Value	High Value	Units	Explanation
Fan24-7StartTime(0-24)	0	24	Hr	Hour of the day when fans start for zones that run 24-7. This is usually equal to 0 since the fans always run for 24-7 operations. This parameter gives the opportunity to change 24-7 operations to more limited operations. **Warning!** For any given building, there may or may not be zones that run in 24-7 mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.
Fan24-7StopTime(0-24)	0	24	Hr	Hour of the day when fans stop for zones that run 24-7. This is usually equal to 24 since the fans always run for 24-7 operations. This parameter gives the opportunity to change 24-7 operations to more limited operations. **Warning!** For any given building, there may or may not be zones that run in 24-7 mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this

				parameter.
FanWD StartTime(0-24)	0	24	Hr	Hour of the day at which fans are turned on during weekdays for all non 24-7 zones in a building. **Warning!** For any given building, there may or may not be zones that run in normal weekday mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.
FanWD StopTime(0-24)	0	24	Hr	Hour of the day at which fans are turned off during weekdays for all non 24-7 zones in a building. **Warning!** For any given building, there may or may not be zones that run in normal weekday mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.
FanWE StartTime(0-24)	0	24	Hr	Hour of the day at which fans are turned on during weekends for all non 24-7 zones in a building. **Warning!** For any given building, there may or may not be zones that run in normal weekend mode. If there are none, then this parameter has no effect. The only way to find out is to search the building input file for this parameter.
FanWE StopTime(0-24)	0	24	Hr	Hour of the day at which fans are turned off during weekends for all non 24-7 zones in a building. **Warning!** For any given building, there may or may not be zones that run in normal weekend mode. If there are none, then this parameter has no effect. The only way to find

out is to search the building input file for this parameter.

Functional Relationships

One to one relationships

How to implement the ECM in a new building:

1. Create the 6 eQUEST parameters above and also create a parameter called “Fan Off Value” = 0.
2. Create three schedules of the forms as seen below using the three pairs of parameters Fan24-7, FanWD and FanWE. An example for 24-7 is provided below (very long)

```
"24-7 ON Fan DSch" = DAY-SCHEDULE-PD
  TYPE          = ON/OFF/FLAG
  VALUES         = (
{If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(1>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   1<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(1>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   1<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
      {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(2>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   2<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(2>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   2<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
      {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(3>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   3<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
```

```
else
 If(3>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   3<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(4>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   4<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(4>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   4<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(5>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   5<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(5>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   5<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(6>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   6<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(6>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   6<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(7>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   7<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(7>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
```

```
    7<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
        {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(8>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    8<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(8>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    8<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
        {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(9>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    9<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(9>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    9<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
        {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(10>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    10<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(10>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    10<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
        {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(11>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    11<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(11>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    11<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
```

```
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(12>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    12<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(12>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    12<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(13>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    13<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(13>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    13<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(14>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    14<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(14>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    14<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(15>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    15<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(15>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    15<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
```

```
 then
 If(16>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    16<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(16>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    16<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(17>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    17<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(17>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    17<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(18>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    18<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(18>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    18<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(19>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
    19<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
else
 If(19>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
    19<=#pa("Fan24-7StopTime(0-24)")) then 1
   else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(20>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
```

```
   20<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(20>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   20<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(21>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   21<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(21>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   21<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(22>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   22<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(22>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   22<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(23>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   23<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
else
 If(23>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   23<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif},
       {If(#pa("Fan24-7StopTime(0-24)") > #pa("Fan24-7StartTime(0-24)"))
 then
 If(24>=#pa("Fan24-7StartTime(0-24)")+1 .AND.
   24<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
```

```
else
 If(24>=#pa("Fan24-7StartTime(0-24)")+1 .OR.
   24<=#pa("Fan24-7StopTime(0-24)")) then 1
  else #pa("Fan Off Value") endif
endif} )
 ..
```

3. Create 2 week (using FanWD and FanWE together) schedules and 2 year schedules for the day schedules created in step 2.
4. In each SYSTEM command place the schedule names in the FAN-SCHEDULE keyword:

```
FAN-SCHEDULE    = "24-7 ON Fan Sch"
```

Reduce Illumination Levels

ECM Description: This ECM seeks to improve building energy performance by reducing the number of light fixtures and/or lamps to prevent excess or unneeded illuminations levels (or foot-candles of light) provided to interior building spaces as well as the building exterior, based on ASHRAE 90.1-2010 light power density standards, thereby reducing building electricity consumption.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
% Reduction to Lamps & FCs	0	100	%	This parameter gives the percent reduction in illumination levels in the building. It is up to the user to ensure that the resultant illumination is acceptable. This can be verified by looking at the “Original LPD” input parameter for the “High Efficiency Lighting Replacement” ECM.

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
LPD Mult	0	100	%	Lighting power density multiplier. This parameter gives the percent reduction in illumination levels in the building. It is up to the user to assure that the resultant illumination is acceptable. This can be verified by

looking at the "Original LPD" input parameter for the "High Efficiency Lighting Replacement" ECM.

Functional Relationships

` LPD Mult ` = 100 - `% Reduction to Lamps & FCs`

How to implement the ECM in a new building:

1. Create the parameter "LPD Mult"
2. In every SPACE command or in the SET-DEFAULT FOR SPACE command include the following key-word:

    LIGHTING-W/AREA = ({#pa("Avg LPD") * #pa("LPD Mult")})

Reduce Plug Loads

ECM Description: This ECM seeks to improve building energy performance by reducing the electricity consumption associated with plug loads, such as computers, monitors, printers, refrigerators, personal space heaters, etc., using Energy Star labeled products, EPEAT registered products, electronics power management policies, advanced power strips, etc., thereby reducing the "equipment power density" of a building.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Plug Load Default Power Density	0	20	W/ft2	The power density due to plug loads. The user must reduce this number using a careful assessment of how much the plug loads can be reduced. This requires understanding what kinds of operations are going on inside each individual building.

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Plugs default W/sf	0	20	W/ft2	The power density due to plug loads. The user must reduce this number using a careful assessment of how

much the plug loads can be reduced. This requires understanding what kinds of operations are going on inside each individual building.

Functional Relationships

One to one

How to implement the ECM in a new building:

1. Create parameter “Plugs default W/sf” and another parameter “EPD Mult”
2. In every SPACE command or in the SET-DEFAULT FOR SPACE command enter the following keyword.

   EQUIPMENT-W/AREA = ({#pa("Plugs default W/sf") * #pa("EPD Mult")})

Seal Vertical Shafts & Stairways

ECM Description: This ECM seeks to improve building energy performance by preventing air buoyancy-induced stack effects occurring in vertical shafts, stairways, and elevators that result in unwanted outdoor air entering the building (vertical infiltration air changes per hour - ACH) causing additional HVAC system load during both heating and cooling seasons. This ECM does not apply to single story buildings.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Vertical Infiltration Air Changes Per Hour	0	2	1/hr	The number of air changes per hour that occur in the building in vertical structures. Usually it is best to either only use this parameter or only use the “Vertical Infiltration Percent Reduction” parameter instead. Using both can be confusing.
Vertical Infiltration Percent Reduction	0	100	%	The percent reduction from “Vertical Infiltration Air Changes Per Hour”

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation

Name	Low Value	High Value	Units	Explanation
Vert Infil ACH	0	2	1/hr	The number of air changes per hour that occur in the building in vertical structures.
Vert Infil % Reduct	0	100	%	The percent reduction from “Vert Infil ACH”

Functional Relationships

One to one

How to implement the ECM in a new building:

1. Create the parameters “Vert Infil ACH” and “Vert Infil % Reduct”
2. In every SPACE which represents a vertical stair way or elevator, insert the following key word-key value pair:
   INF-FLOW/AREA =
   {#pa("Vert Infil ACH") * local("VOLUME") / local("AREA") / 60
   * (1 - #pa("Vert Infil % Reduct"))
   }

Supply Air Temperature Reset

ECM Description: This ECM seeks to improve building energy performance by implementing a supply air temperature (SAT) reset control strategy to increase or decrease the supply air temperature setting in variable-air-volume systems based on indoor/outdoor air temperatures that ensures occupant comfort level while reducing heating and cooling load demands on the HVAC system. The ECM covers both heating and cooling supply air temperature resets.

For cooling, a SAT reset is a control strategy that takes advantage of times when cooling demand is active but well below the capacity of the air handling units. In such cases, the supply air temperature can be increased (for example from 55F to 65F). Doing this often increases the comfort of occupants (i.e. an occupant stuck in a cold stream of 55F air will be cold whereas a 65F stream would be more comfortable), reduces reheating, and increases the number of economizing hours. These benefits are counteracted with an increase in fan operation.

This ECM has two reset type parameters that both have three options. When equal to 0 both the cooling and heating reset parameters are set to constant values so that there is no supply air temperature reset. If set to 1, these parameters create a supply air temperature reset for cooling or heating based on outdoor temperature as seen in Figure 45 and Figure 46. The ECM uses a mixture law between existing values and new values to derive an average SAT reset schedule. To avoid confusion it is often best to give “%New Cooling SAT MinT”, “%New Cooling SAT Reset MaxT”, “%New Heating SAT

MaxT" and "%New Heating SAT Reset MinT" values of 100% so that the "new" value parameters dictate the SAT reset profile. When the reset type parameters are equal to 2, a different type of reset which is controlled by the warmest zone for cooling and the coolest zone for heating is used. For this case the minimum temperature for heating SAT reset and maximum temperature for cooling SAT reset have no effect.

For the current ECM, the outdoor air temperatures at which the reset maximum temperature and reset minimum temperatures saturate are hardwired into the building model as 60F and 80F for cooling and as 30F and 70F for heating.

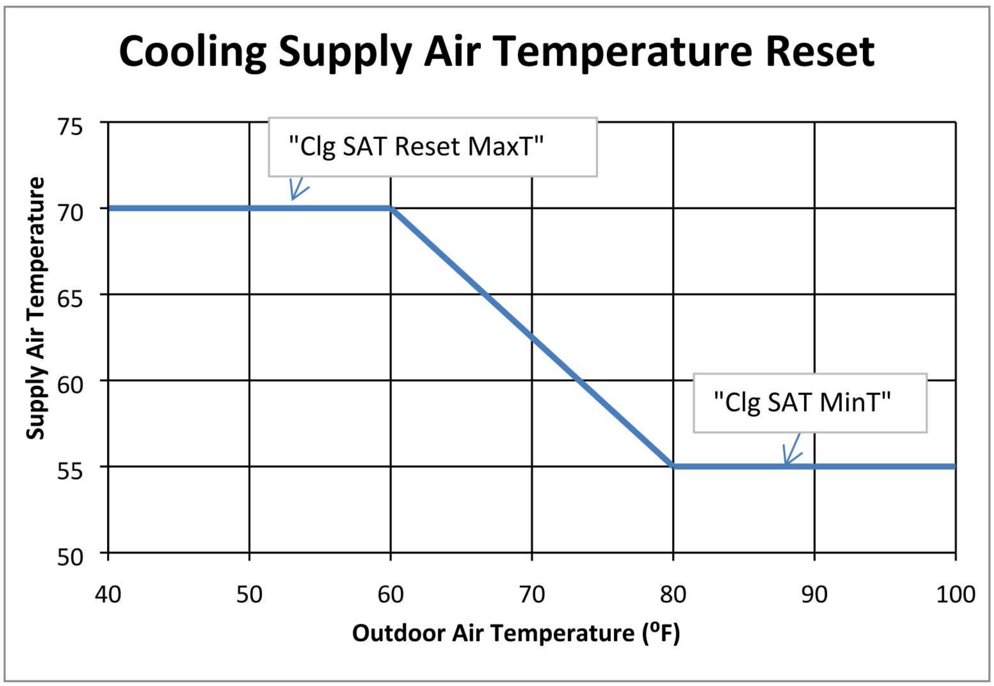


Figure 45. Example cooling SAT set-point as a function of outdoor air temperature

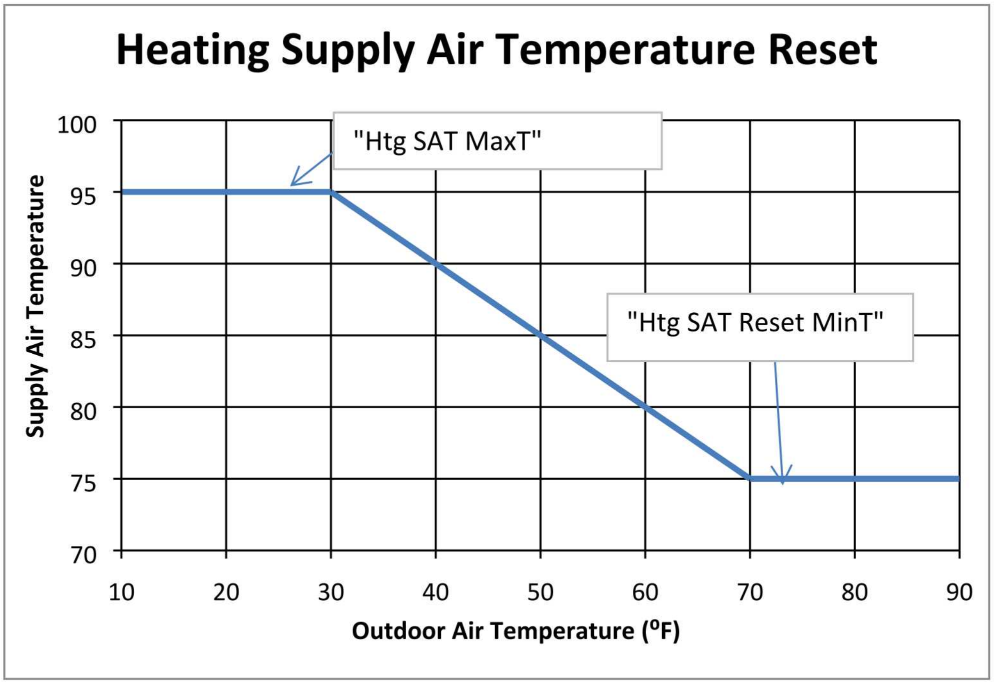


Figure 46. Example heating SAT set-point as a function of outdoor air temperature

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
%New Cooling SAT MinT	0	100	%	Percentage of zones to apply a new minimum cooling temperature to. This parameter is used in a mixture law to calculate an average cooling minimum temperature ("Clg SAT MinT")
Existing Cooling SAT MinT	45	85	Fahrenheit	For the existing system, the minimum air temperature delivered to zones in the building. **(this parameter does not need to be changed)**
New Cooling SAT MinT	45	85	Fahrenheit	The new minimum temperature delivered to zones in the building. This new value is only delivered to "%New Cooling SAT MinT" of the zones
%New Cooling SAT	0	100	%	Percentage of zones to apply a new

Name	Low Value	High Value	Units	Explanation
Reset MaxT				maximum SAT reset temperature to. This parameter is used in a mixture law to calculate an average cooling maximum SAT reset temperature ("Clg SAT Reset MaxT")
Exist Cooling SAT Reset MaxT	20	90	Fahrenheit	Existing maximum cool air supply temperature allowed for a SAT Reset. Specifies the maximum temperature to which the supply air temperature resets during cooling **(this parameter does not need to be changed)**
New Cooling SAT Reset MaxT	45	85	Fahrenheit	The new maximum SAT delivered to zones. This parameter is used in a mixture law to calculate an average cooling maximum SAT reset temperature ("Clg SAT Reset MaxT")
New Cooling SAT Reset Type	0	2	none	0 = CONSTANT, 1 = RESET, 2 = WARMEST CONSTANT = Sets heating supply and/or cooling supply air temperature to a fixed value. Values should then be entered for keywords HEAT-SET-T and/or COOL-SET-T, respectively RESET = Specifies use of HEAT-RESET-SCH or COOL-RESET-SCH for control of heating and/or cooling air supply temperature, based on outdoor air temperature. WARMEST = Sets the cooling coil

Name	Low Value	High Value	Units	Explanation
				(cold deck) temperature each hour to adequately cool the ZONE with the highest temperature. The limits on the supply air temperature are governed by coil capacities, cooling schedules, and MIN-SUPPLY-T.
%New Heating SAT MaxT	0	100	%	Percent of zones which will have a new maximum heating SAT. This parameter is used in a mixture law to calculate an average heating maximum temperature ("Htg SAT MaxT")
Exist Heating SAT MaxT	45	85	Fahrenheit	Heating SAT maximum temperature for existing control. **(this parameter does not need to be changed)**
New Heating SAT MaxT	45	85	Fahrenheit	Heating SAT maximum temperature for new control
%New Heating SAT Reset MinT	0	100	%	Percent of zones which will have a new minimum heating SAT
Exist Heating SAT Reset MinT	45	85	Fahrenheit	Existing SAT Reset heating minimum temperature **(this parameter does not need to be changed)**
New Heating SAT Reset MinT	45	85	Fahrenheit	New SAT Reset heating minimum temperature
New Heating SAT Reset Type	0	2	Fahrenheit	0 = CONSTANT, 1 = RESET, 2 = COLDEST CONSTANT = The supply air temperature is held nearly constant at HEAT-SET-T; it can change a few degrees due to fan heat and throttling range of the temperature controller.

Name	Low Value	High Value	Units	Explanation
				COLDEST = Sets the heating coil (hot deck) temperature each hour to adequately heat the zone with the lowest temperature. The limits on the supply air temperature are governed by coil capacities, heating schedules, and MAX-SUPPLY-T. WARNING: If using the COLDEST or WARMEST options in conjunction with a variable air volume system, there are two actions within the throttling range. To reflect reality and to prevent instability in the simulation, THROTTLING-RANGE should be increased to 4-6F (2-3K). RESET = Specifies use of HEAT-RESET-SCH for control of heating air supply temperature based on outdoor air temperature.

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
Clg SAT MinT	40	80	Fahrenheit	Average minimum cooling SAT across all AHUs (air handling units), The lowest allowable temperature for air entering the ZONE(s), that is, the lowest allowed diffuser temperature. Doe2.2 uses this temperature to determine design supply air flow rate.
Clg SAT Reset MaxT	40	90	Fahrenheit	Average maximum temperature for cooling SAT reset.

Name	Low Value	High Value	Units	Explanation
Clg SAT Reset Type	0	2	None	0 = CONSTANT, 1 = RESET, 2 = WARMEST CONSTANT = Sets heating supply and/or cooling supply air temperature to a fixed value. Values should then be entered for keywords HEAT-SET-T and/or COOL-SET-T, respectively RESET = Specifies use of HEAT-RESET-SCH or COOL-RESET-SCH for control of heating and/or cooling air supply temperature, based on outdoor air temperature. WARMEST = Sets the cooling coil (cold deck) temperature each hour to adequately cool the ZONE with the highest temperature. The limits on the supply air temperature are governed by coil capacities, cooling schedules, and MIN-SUPPLY-T.
Htg SAT MaxT	40	80	Fahrenheit	Average maximum temperature for heating SAT reset.
Htg SAT Reset MinT	40	80	Fahrenheit	Average minimum temperature for heating SAT reset.
Htg SAT Reset Type	0	2	None	0 = CONSTANT, 1 = RESET, 2 = COLDEST CONSTANT = The supply air temperature is held nearly constant at HEAT-SET-T; it can

Name	Low Value	High Value	Units	Explanation
				change a few degrees due to fan heat and throttling range of the temperature controller. COLDEST = Sets the heating coil (hot deck) temperature each hour to adequately heat the zone with the lowest temperature. The limits on the supply air temperature are governed by coil capacities, heating schedules, and MAX-SUPPLY-T. WARNING: If using the COLDEST or WARMEST options in conjunction with a variable air volume system, there are two actions within the throttling range. To reflect reality and to prevent instability in the simulation, THROTTLING-RANGE should be increased to 4-6F (2-3K). RESET = Specifies use of HEAT-RESET-SCH for control of heating air supply temperature based on outdoor air temperature.

Functional Relationships

1. ` Clg SAT MinT ` = `Exist Cooling SAT MinT` * (1 - `%New Cooling SAT MinT`) + `New Cooling SAT MinT` * `%New Cooling SAT MinT`
2. ` Clg SAT Reset MaxT ` = `Exist Cooling SAT Reset MaxT`*(1-`%New Cooling SAT Reset MaxT`)+`New Cooling SAT Reset MaxT`*`%New Cooling SAT Reset MaxT`
3. `Clg SAT Reset Type` = `New Cooling SAT Reset Type`
4. `Htg SAT MaxT` = `Exist Heating SAT MaxT` * (1 - `%New Heating SAT MaxT`) + `New Heating SAT MaxT` * `%New Heating SAT MaxT`

5. `Htg SAT Reset MinT` = `Exist Heating SAT Reset MinT`*(1-`%New Heating SAT Reset MinT`)+`New Heating SAT Reset MinT` *`%New Heating SAT Reset MinT`

6. `Htg SAT Reset Type` = `New Heating SAT Reset Type`

<u>How to implement the ECM in a new building:</u>

1. Create all of the parameters in the table above for eQUEST read only parameters.
2. For every SYSTEM command which involve a cooling system insert the following keyword-key-value pairs:

```
COOL-RESET-SCH   =
{if(#pa("Clg SAT Reset Type") = 1) then
  #si("Clg SAT OA Reset Sch","SYSTEM","COOL-RESET-SCH")
 else
  unused
 endif}
  COOL-CONTROL    =
{if(#pa("Clg SAT Reset Type") = 1) then
 #si("RESET","SYSTEM","COOL-CONTROL")
else if(#pa("Clg SAT Reset Type") = 2) then
 #si("WARMEST","SYSTEM","COOL-CONTROL")
else
 #si("CONSTANT","SYSTEM","COOL-CONTROL")
endif endif}
  COOL-MAX-RESET-T = {#pa("Clg SAT Reset MaxT")}
  MIN-SUPPLY-T     = {#pa("Clg SAT MinT")}
  MAX-SUPPLY-T     = {#pa("Htg SAT MaxT")
  HEAT-RESET-SCH   =
{if(#pa("Htg SAT Reset Type") = 1) then
  #si("Htg SAT OA Reset Sch","SYSTEM","HEAT-RESET-SCH")
 else
  unused
 endif}
  HEAT-CONTROL    =
{if(#pa("Htg SAT Reset Type") = 1) then
 #si("RESET","SYSTEM","HEAT-CONTROL")
else if(#pa("Htg SAT Reset Type") = 2) then
 #si("COLDEST","SYSTEM","HEAT-CONTROL")
else
 #si("CONSTANT","SYSTEM","HEAT-CONTROL")
endif endif
```

```
}
```

3. Create yearly schedules named “Clg SAT OA Reset Sch” and “Htg SAT OA Reset Sch”

```
"Clg SAT OA Reset Sch" = SCHEDULE-PD
  TYPE          = RESET-TEMP
  MONTH         = ( 12 )
  DAY           = ( 31 )
  WEEK-SCHEDULES  = ( "Clg SAT OA Reset WSch" )
 ..

"Htg SAT OA Reset Sch" = SCHEDULE-PD
  TYPE          = RESET-TEMP
  MONTH         = ( 12 )
  DAY           = ( 31 )
  WEEK-SCHEDULES  = ( "Htg SAT OA Reset WSch" )
  ..
```

4. Create weekly schedules “Clg SAT OA Reset WSch” and "Htg SAT OA Reset WSch"

```
"Clg SAT OA Reset WSch" = WEEK-SCHEDULE-PD
  TYPE          = RESET-TEMP
  DAY-SCHEDULES   = ( "Clg SAT OA Reset DSch" )
 ..

"Htg SAT OA Reset WSch" = WEEK-SCHEDULE-PD
  TYPE          = RESET-TEMP
  DAY-SCHEDULES   = ( "Htg SAT OA Reset DSch" )
  ..
```

5. Create daily schedules “Clg SAT OA Reset DSch” and “Htg SAT OA Reset DSch”

```
"Clg SAT OA Reset DSch" = DAY-SCHEDULE-PD
  TYPE          = RESET-TEMP
  OUTSIDE-HI     = 80
  OUTSIDE-LO      = 60
  SUPPLY-HI      = {#pa("Clg SAT Reset MaxT")}
  SUPPLY-LO       = {#pa("Clg SAT MinT")}
..

"Htg SAT OA Reset DSch" = DAY-SCHEDULE-PD
  TYPE          = RESET-TEMP
```

```
OUTSIDE-HI      = 70
OUTSIDE-LO      = 30
SUPPLY-HI       = {#pa("Htg SAT MaxT")}
SUPPLY-LO       = {#pa("Htg SAT Reset MinT")}
..
```

Thermostat Management (temperature set-back and occupancy set-point)

ECM Description: This ECM allows the user to change the temperature set-points of a building to reduce energy consumption. The user needs to be aware of the implications for occupant comfort in setting the parameters involved. Thermostat parameters are present for 24/7 spaces of a building and normal operations spaces. If a building does not have any 24/7 or is all 24/7, then half of the parameters will not have any effect since they are not represented by any spaces. Parameters are also present for occupied and unoccupied periods.

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
24/7 Cooling Occupied Thermostat Setpoint	50	100	Fahrenheit	Cooling Set point for all spaces that have 24/7 operations during occupied hours
24/7 Cooling UnOccupied Thermostat Setpoint	50	100	Fahrenheit	Cooling Set point for all spaces that have 24/7 operations during unoccupied hours
24/7 Heating Occupied Thermostat Setpoint	50	100	Fahrenheit	Heating Set point for all spaces that have 24/7 operations during occupied hours
24/7 Heating Unoccupied Thermostat Setpoint	50	100	Fahrenheit	Heating Set point for all spaces that have 24/7 operations during unoccupied hours
Cooling Occupied Thermostat Setpoint	50	100	Fahrenheit	Cooling Set point for normal operations spaces during occupied hours
Cooling Unoccupied Thermostat Setpoint	50	100	Fahrenheit	Cooling Set point for normal operations spaces during unoccupied hours
Heating Occupied Thermostat Setpoint	50	100	Fahrenheit	Heating Set point for normal operations spaces during occupied

Name	Low Value	High Value	Units	Explanation
				hours
Heating Unoccupied Thermostat Setpoint	50	100	Fahrenheit	Heating Set point for normal operations spaces during unoccupied hours

Parameters In eQUEST/Doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
24/7 Cool Occ T-Stat Setpt	50	100	Fahrenheit	Cooling Set point for all spaces that have 24/7 operations during occupied hours
24/7 Cool UnOcc T-Stat Setpt	50	100	Fahrenheit	Cooling Set point for all spaces that have 24/7 operations during unoccupied hours
24/7 Heat Occ T-Stat Setpt	50	100	Fahrenheit	Heating Set point for all spaces that have 24/7 operations during occupied hours
24/7 Heat UnOcc T-Stat Setpt	50	100	Fahrenheit	Heating Set point for all spaces that have 24/7 operations during unoccupied hours
Cooling Occ T-Stat Setpoint	50	100	Fahrenheit	Cooling Set point for normal operations spaces during occupied hours
Cooling UnOcc T-Stat Setpoint	50	100	Fahrenheit	Cooling Set point for normal operations spaces during unoccupied hours
Heating Occ T-Stat Setpoint	50	100	Fahrenheit	Heating Set point for normal operations spaces during occupied hours
Heating UnOcc T-Stat Setpoint	50	100	Fahrenheit	Heating Set point for normal operations spaces during unoccupied hours

Functional Relationships

One to one between the parameters in the tables above

How to implement the ECM in a new building: (This procedure is very complex and needs updating)

1. Create all the parameters for this ECM and for the “VAV Box Occupancy Sensors” and “Limit Personal Space Heater Use” ECM’s.
2. Create day schedules as follows for 24/7 operations (for workday/weekend and heating/cooling)

```
"Heat T-Stat (24/7) DSch2" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES        = ( {#pa("24/7 Heat UnOcc T-Stat Setpt")}, &D, &D, &D, &D,
      &D, {#pa("24/7 Heat Occ T-Stat Setpt")}, &D, &D, &D, &D, &D, &D, &D,
      &D, &D, &D, &D, {#pa("24/7 Heat UnOcc T-Stat Setpt")} )
  ..
"Cool T-Stat (24/7) DSch2" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES        = ( {#pa("24/7 Cool UnOcc T-Stat Setpt")}, &D, &D, &D, &D,
      &D, {#pa("24/7 Cool Occ T-Stat Setpt")}, &D, &D, &D, &D, &D, &D, &D,
      &D, &D, &D, &D, {#pa("24/7 Cool UnOcc T-Stat Setpt")} )
  ..
"Heat T-Stat (24/7) WEDSch2" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES        = ( {#pa("24/7 Heat UnOcc T-Stat Setpt")} )
  ..
"Cool T-Stat (24/7) WEDSch2" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES        = ( {#pa("24/7 Cool UnOcc T-Stat Setpt")} )
  ..
```

3. Create weekly schedules for 24/7 operations

```
"Heat T-Stat (24/7) WSch2" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES    = ( "Heat T-Stat (24/7) DSch2", &D, &D, &D, &D,
      "Heat T-Stat (24/7) WEDSch2" )
  ..
"Cool T-Stat (24/7) WSch2" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES    = ( "Cool T-Stat (24/7) DSch2", &D, &D, &D, &D,
      "Cool T-Stat (24/7) WEDSch2" )
  ..
```

4. Create yearly schedules for 24/7 operations

```
"Heat T-Stat (24/7) Sch2" = SCHEDULE-PD
  TYPE          = TEMPERATURE
  MONTH         = ( 12 )
  DAY           = ( 31 )
  WEEK-SCHEDULES   = ( "Heat T-Stat (24/7) WSch2" )
  ..
"Cool T-Stat (24/7) Sch2" = SCHEDULE-PD
  TYPE          = TEMPERATURE
  MONTH         = ( 12 )
  DAY           = ( 31 )
  WEEK-SCHEDULES   = ( "Cool T-Stat (24/7) WSch2" )
  ..
```

5. Create day schedules for normal operations as follows. These schedules have additional parameters which are part of a different ECM

```
"Occ Heat T-Stat WD DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES        = ( {#pa("Heating UnOcc T-Stat Setpoint")}, &D, &D, &D,
      &D, &D, {#pa("Heating Occ T-Stat Setpoint")}, &D, &D, &D, &D, &D,
      &D, &D, &D, &D, &D, &D, {#pa("Heating UnOcc T-Stat Setpoint")} )
  ..
"Occ Heat T-Stat WEH DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES        = ( {#pa("Heating UnOcc T-Stat Setpoint")} )
  ..
"Occ Cool T-Stat WD DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES        = ( {#pa("Cooling UnOcc T-Stat Setpoint")}, &D, &D, &D,
      &D, &D, {#pa("Cooling Occ T-Stat Setpoint")}, &D, &D, &D, &D, &D,
      &D, &D, &D, &D, &D, &D, {#pa("Cooling UnOcc T-Stat Setpoint")} )
  ..
"Occ Cool T-Stat WEH DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES        = ( {#pa("Cooling UnOcc T-Stat Setpoint")} )
  ..

"UnOcc Cool T-Stat WD DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
```

```
  VALUES          = ( {#pa("Cooling UnOcc T-Stat Setpoint")}, &D, &D, &D,
     &D, &D, &D,
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 7a-8a")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 8a-9a")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 9a-10a")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 10a-11a")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 11a-12p")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 12p-1p")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 1p-2p")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 2p-3p")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 3p-4p")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 4p-5p")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 5p-6p")) then
      #pa("Cooling UnOcc T-Stat Setpoint") else
   #pa("Cooling Occ T-Stat Setpoint") endif},
     {#pa("Cooling UnOcc T-Stat Setpoint")} )
  ..
"UnOcc Cool T-Stat WEH DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES          = ( {#pa("Cooling UnOcc T-Stat Setpoint")} )
  ..
"UnOcc Heat T-Stat WD DSch" = DAY-SCHEDULE-PD
```

```
  TYPE          = TEMPERATURE
  VALUES          = ( {#pa("Heating UnOcc T-Stat Setpoint")}, &D, &D, &D,
      &D, &D, &D,
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 7a-8a")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 8a-9a")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 9a-10a")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 10a-11a")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 11a-12p")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 12p-1p")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 1p-2p")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 2p-3p")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 3p-4p")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 4p-5p")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {if(#pa("#Hrs/Day Unoc")>=#pa("UnOc#HrsRank 5p-6p")) then
       #pa("Heating UnOcc T-Stat Setpoint") else
   #pa("Heating Occ T-Stat Setpoint") endif},
      {#pa("Heating UnOcc T-Stat Setpoint")} )
  ..
"UnOcc Heat T-Stat WEH DSch" = DAY-SCHEDULE-PD
  TYPE          = TEMPERATURE
  VALUES          = ( {#pa("Heating UnOcc T-Stat Setpoint")} )
  ..
```

6. Create week schedules as follows

```
"UnOcc Cool T-Stat WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "UnOcc Cool T-Stat WD DSch", &D, &D, &D, &D,
      "UnOcc Cool T-Stat WEH DSch" )
  ..
"UnOcc Cool T-Stat Hol WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "UnOcc Cool T-Stat WD DSch" )
  ..
"UnOcc Heat T-Stat WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "UnOcc Heat T-Stat WD DSch", &D, &D, &D, &D,
      "UnOcc Heat T-Stat WEH DSch" )
  ..
"UnOcc Heat T-Stat Hol WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "UnOcc Heat T-Stat WD DSch" )
  ..
"Occ Heat T-Stat Normal WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "Occ Heat T-Stat WD DSch", &D, &D, &D, &D,
      "Occ Heat T-Stat WEH DSch" )
  ..
"Heat T-Stat Holiday WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "Occ Heat T-Stat WEH DSch" )
  ..
"Occ Cool T-Stat Normal WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "Occ Cool T-Stat WD DSch", &D, &D, &D, &D,
      "Occ Cool T-Stat WEH DSch" )
  ..
"Cool T-Stat Holiday WSch" = WEEK-SCHEDULE-PD
  TYPE          = TEMPERATURE
  DAY-SCHEDULES   = ( "Occ Cool T-Stat WEH DSch" )
  ..
```

7. Create yearly schedules (these assume the Sandia winter shutdown is occurring, which is probably not the case for most sites)

```
"Occ Heat T-Stat Sch" = SCHEDULE-PD
  TYPE          = TEMPERATURE
  MONTH         = ( 12, 12 )
  DAY           = ( 23, 31 )
  WEEK-SCHEDULES   = ( "Occ Heat T-Stat Normal WSch",
      "Heat T-Stat Holiday WSch" )
  ..
"Occ Cool T-Stat Sch" = SCHEDULE-PD
  TYPE          = TEMPERATURE
  MONTH         = ( 12, 12 )
  DAY           = ( 23, 31 )
  WEEK-SCHEDULES   = ( "Occ Cool T-Stat Normal WSch",
      "Cool T-Stat Holiday WSch" )
  ..

"UnOcc Cool T-Stat Sch" = SCHEDULE-PD
  TYPE          = TEMPERATURE
  MONTH         = ( 12, 12 )
  DAY           = ( 23, 31 )
  WEEK-SCHEDULES   = ( "UnOcc Cool T-Stat WSch",
      "UnOcc Cool T-Stat Hol WSch" )
  ..
"UnOcc Heat T-Stat Sch" = SCHEDULE-PD
  TYPE          = TEMPERATURE
  MONTH         = ( 12, 12 )
  DAY           = ( 23, 31 )
  WEEK-SCHEDULES   = ( "UnOcc Heat T-Stat WSch",
      "UnOcc Heat T-Stat Hol WSch" )
```

8. In the SET-DEFAULT FOR ZONE command or for every ZONE (necessary for CUBs) enter the following. In this programming the DESIGN-COOL-T is being used to store information about zones in the model to indicate whether a zone is occupied or unoccupied in the VAV Box Occupancy Sensors ECM. The non-integer portion of DESIGN-COOL-T must be less than “Frac Offices Unoc” in order for a specific zone to receive the special “unoccupied during occupied hours schedule.” It is expected that the modeler has assigned values that cause the distribution of offices to track the “Frac Offices Unoc.” The first office to be unoccupied is given a value of equal to the fraction of area it occupies, then the next office chosen is given a value equal to its own area fraction plus that of the 1st office. This continues to the last office. If this is not done, then the “Frac Offices Unoc” will not work correctly. The DESIGN-HEAT-T for each ZONE is being used to indicate whether a special schedule with space heaters is being used. The non-integer

portion must be less than “Frac Ofcs w SpcHtrs” in order for the special space heater schedule to be used.

```
  HEAT-TEMP-SCH    =
{ If(local("DESIGN-COOL-T")-INT(local("DESIGN-COOL-T"))=0) then

 If(local("DESIGN-HEAT-T")-INT(local("DESIGN-HEAT-T"))=0) then
  #SI("Occ Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
 else
  If(#pa("Frac Ofcs w SpcHtrs") >= INT((local("DESIGN-HEAT-T")-
     INT(local("DESIGN-HEAT-T")))*100)/100 ) then
   #SI("SpHtr Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
  else
   #SI("Occ Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
  endif
 endif

else

 If(#pa("Frac Offices Unoc") >= INT((local("DESIGN-COOL-T")-
   INT(local("DESIGN-COOL-T")))*100)/100 ) then

   If(local("DESIGN-HEAT-T")-INT(local("DESIGN-HEAT-T"))=0) then
    #SI("UnOcc Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
   else
    If(#pa("Frac Ofcs w SpcHtrs") >= INT((local("DESIGN-HEAT-T")-
      INT(local("DESIGN-HEAT-T")))*100)/100 ) then
     #SI("SpHtr Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
    else
     #SI("UnOcc Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
    endif
   endif

 else
  If(local("DESIGN-HEAT-T")-INT(local("DESIGN-HEAT-T"))=0) then
   #SI("Occ Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
  else
   If(#pa("Frac Ofcs w SpcHtrs") >= INT((local("DESIGN-HEAT-T")-
     INT(local("DESIGN-HEAT-T")))*100)/100 ) then
    #SI("SpHtr Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
   else
    #SI("Occ Heat T-Stat Sch","ZONE","HEAT-TEMP-SCH")
```

```
    endif
   endif
  endif

 endif }
   DESIGN-COOL-T    = {#pa("Cooling Occ T-Stat Setpoint")-1}
   COOL-TEMP-SCH    =
{If(local("DESIGN-COOL-T")-INT(local("DESIGN-COOL-T"))=0) then
 #SI("Occ Cool T-Stat Sch","ZONE","COOL-TEMP-SCH") else

 If(#pa("Frac Offices Unoc") >= INT((local("DESIGN-COOL-T")-
   INT(local("DESIGN-COOL-T")))*100)/100 ) then

 #SI("UnOcc Cool T-Stat Sch","ZONE","COOL-TEMP-SCH") else
 #SI("Occ Cool T-Stat Sch","ZONE","COOL-TEMP-SCH")
 endif endif}
```

9.

Variable Air Volume (VAV) Box Occupancy Sensors

ECM Description: This ECM seeks to improve building energy performance by applying occupancy sensors to control (on/off) building zone level variable-air-volume terminal units during standard operating hours of the building, thereby reducing energy demands on the building HVAC system when a specific zone or zones of the building are unoccupied during the day.

The user first inputs a “Number of Hours Offices Unoccupied.” This can be a value from 0 to 11 to indicate how many hours of normally occupied time are actually unoccupied. For example a value of 1 can be entered to approximate the presence of lunchtime. A higher value of 2 or 3 may approximate that staff are spending more time in labs than their offices. The second input “Fraction of Office Zones Unoccupied” is used to estimate how many offices are left unoccupied. For example, a value of 0.5 would indicate that half the offices in the building are unoccupied during the hours indicated as unoccupied. The next 11 parameters are used to designate which hours are actually unoccupied. In Table 2, an example distribution is given. If “Number of Hours of Offices Unoccupied” is equal to 1, then “Fraction of Office Zones Unoccupied” is unoccupied from 12-1pm. If it is equal to 3 then offices are unoccupied for 12-1pm, 7-8am, and 5-6pm. Usually the rank order does not need to changed unless the user has collected some specific occupancy data.

Table 2. Example rank distribution

Rank 7-8am	2
Rank 8-9am	6

Rank 9-10am	8
Rank 10-11am	10
Rank 11am-12pm	11
Rank 12-1pm	1
Rank 1-2pm	9
Rank 2-3pm	7
Rank 3-4pm	5
Rank 4-5pm	4
Rank 5-6pm	3

IX Input Parameters:

Name	Low Value	High Value	Units	Explanation
Number of Hours Offices Unoccupied	0	11	none	The number of hours for which offices are unoccupied during normally occupied hours. This only determines the number of hours. Which hours are unoccupied is determined by the rank parameters
Fraction of Office Zones Unoccupied	0	1	fraction	This is the fraction of office zones which are unoccupied during the hours indicated by the "Number of Hours Offices Unoccupied" and the rank parameters
Rank 7-8am	1	11	None	See ECM Description/Table 2
Rank 8-9am	1	11	None	See ECM Description/Table 2
Rank 9-10am	1	11	None	See ECM Description/Table 2
Rank 10-11am	1	11	None	See ECM Description/Table 2
Rank 11am-12pm	1	11	None	See ECM Description/Table 2

Rank 12-1pm	1	11	None	See ECM Description/Table 2
Rank 1-2pm	1	11	None	See ECM Description/Table 2
Rank 2-3pm	1	11	None	See ECM Description/Table 2
Rank 3-4pm	1	11	None	See ECM Description/Table 2
Rank 4-5pm	1	11	None	See ECM Description/Table 2
Rank 5-6pm	1	11	None	See ECM Description/Table 2

Parameters In eQUEST/doe2.2 (read only):

Name	Low Value	High Value	Units	Explanation
#Hrs/Day Unoc	0	11	none	The number of hours for which offices are unoccupied during normally occupied hours. This only determines the number of hours. Which hours are unoccupied is determined by the rank parameters
Frac Offices Unoc	0	1	fraction	This is the fraction of office zones that are unoccupied during the hours indicated by the “Number of Hours Offices Unoccupied” and the rank parameters
UnOc#HrsRank 7a-8a	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 8a-9a	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 9a-10a	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 10a-11a	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 11a-12p	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 12p-1p	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 1p-2p	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 2p-3p	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 3p-4p	1	11	None	See ECM Description/Table 2

Name	Low Value	High Value	Units	Explanation
UnOc#HrsRank 4p-5p	1	11	None	See ECM Description/Table 2
UnOc#HrsRank 5p-6p	1	11	None	See ECM Description/Table 2

Functional Relationships

All relationships are one to one relationships in the table above.

How to implement the ECM in a new building:

Implementing the Thermostat Management ECM will simultaneously implement this ECM.

Appendix B User Interface Tutorial

To begin using IX, open the IX User Interface by double clicking "IXUserInterface2_5.xlsm." Be sure to "Enable Content" as seen in Figure 47 so that Visual Basic for Application (VBA) functions can run.

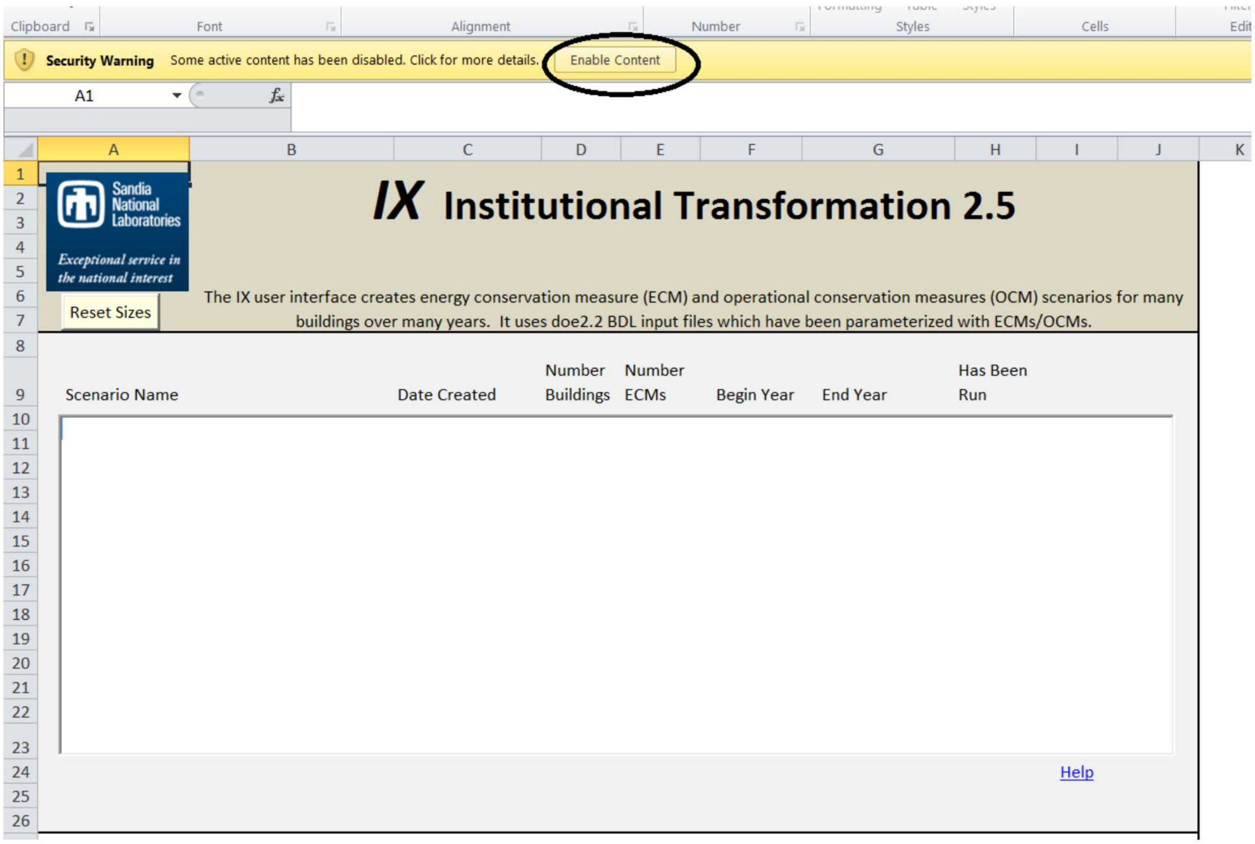


Figure 47. Enable content for VBA

The application always opens in scenario management. When no scenarios have been created, a right click within the list box provides the option to create a new scenario as seen in Figure 48. Clicking on "New" prompts the user to enter a scenario name as seen in Figure 49. Only alpha numeric characters (no spaces) and "_" are allowed for scenario names. Be extra careful in choosing a name because there is no way to rename afterwards. Click on "OK." This creates an empty scenario that needs to have data input into it.

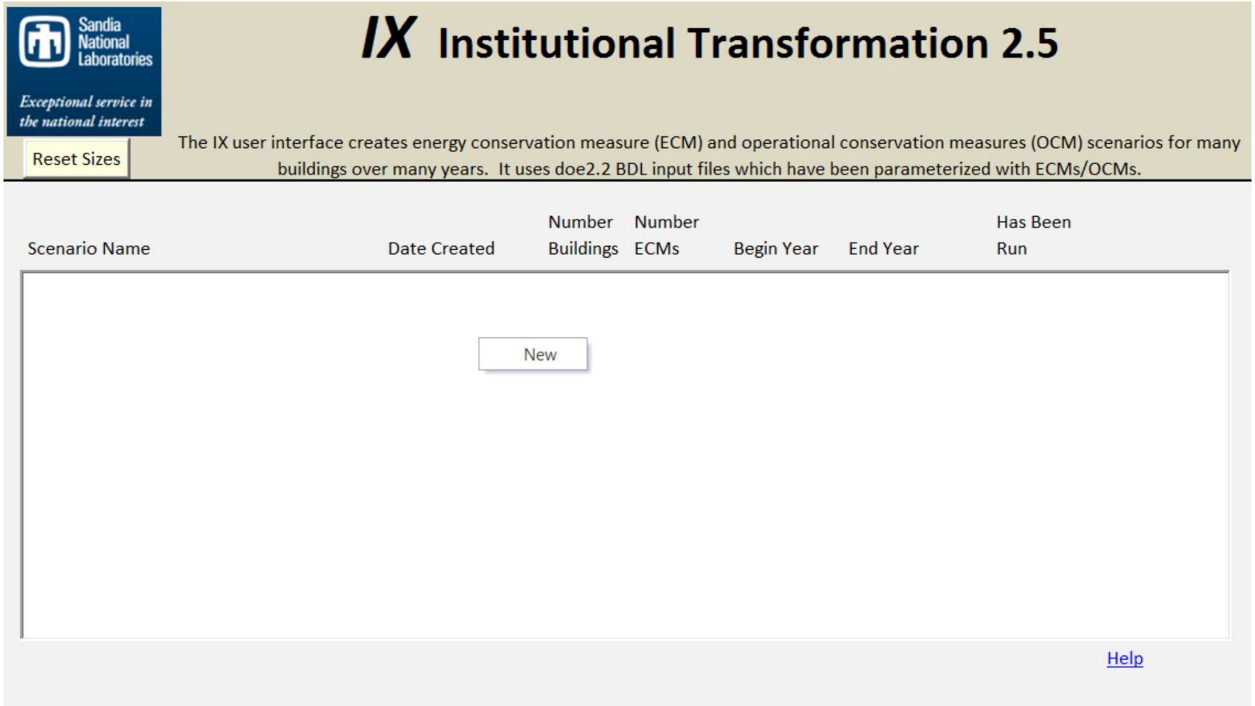


Figure 48. Creating a new scenario

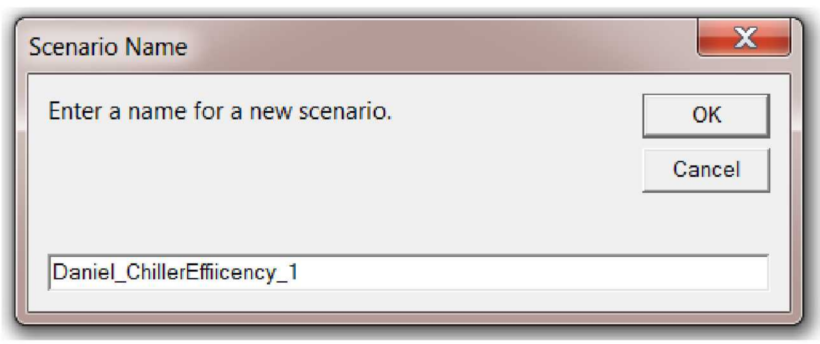


Figure 49. Enter a new scenario name

Right click on the empty scenario and a number of new options are available as shown in Figure 50. Select “Edit.” Selection of “Run” or “Results” at this point is not meaningful because there is not any input in the scenario. The “Delete” option will erase the scenario and the “Copy” option can be used to copy an entire scenario to a new name.

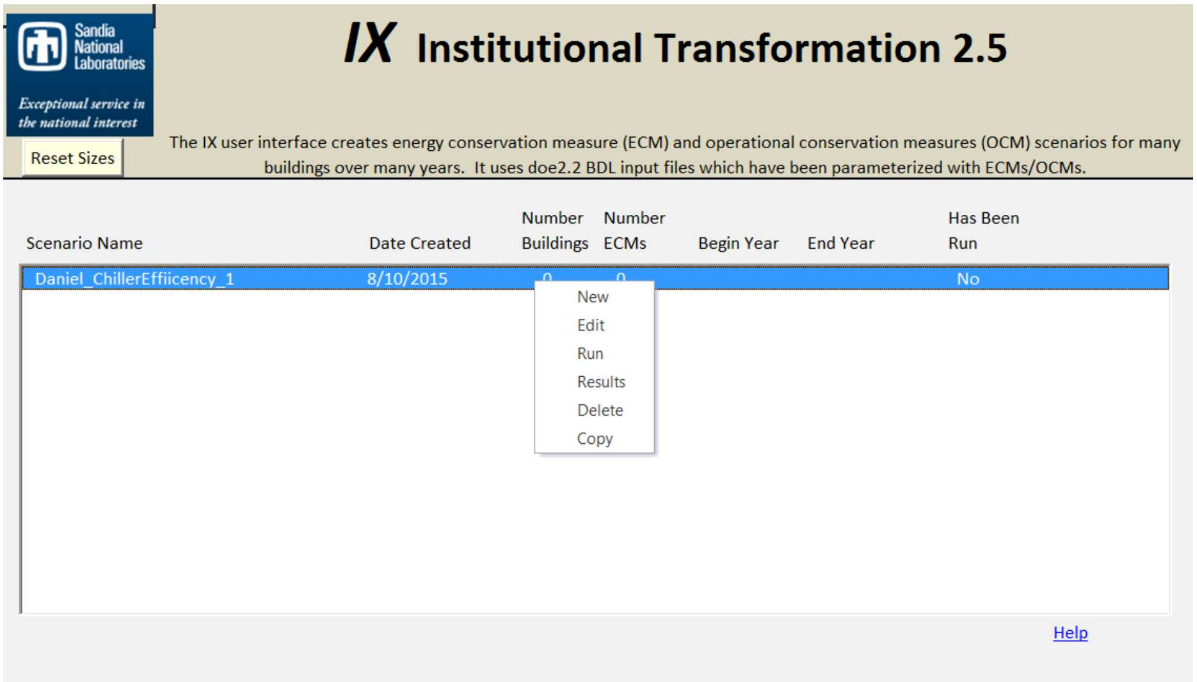


Figure 50. Right click on an existing scenario

The Input page shows after selecting "Edit." This page is used to select the ECMs, Years, and Buildings to apply to your scenario and to enter the required data for parameters within each ECM. The uppermost combo box in Figure 51 can be used to switch between scenarios. We will select the CUB Chiller Efficiency ECM to Add to our scenario as seen in Figure 52. Next select a begin year of 2015 and an end year of 2016 as seen in Figure 53.

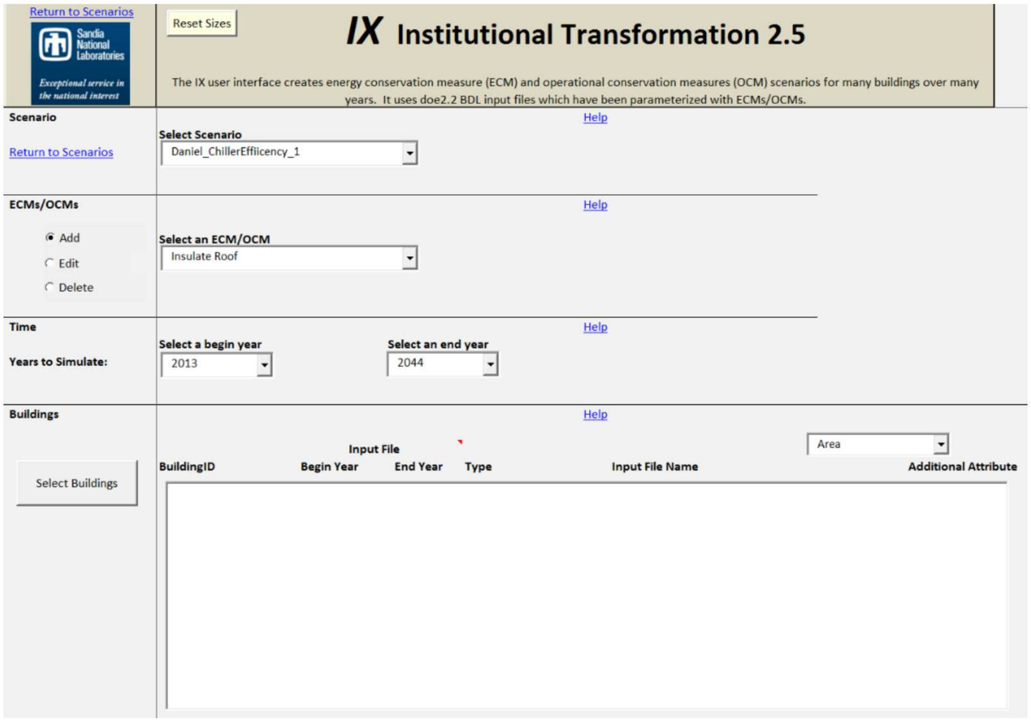


Figure 51. Edit page upper half

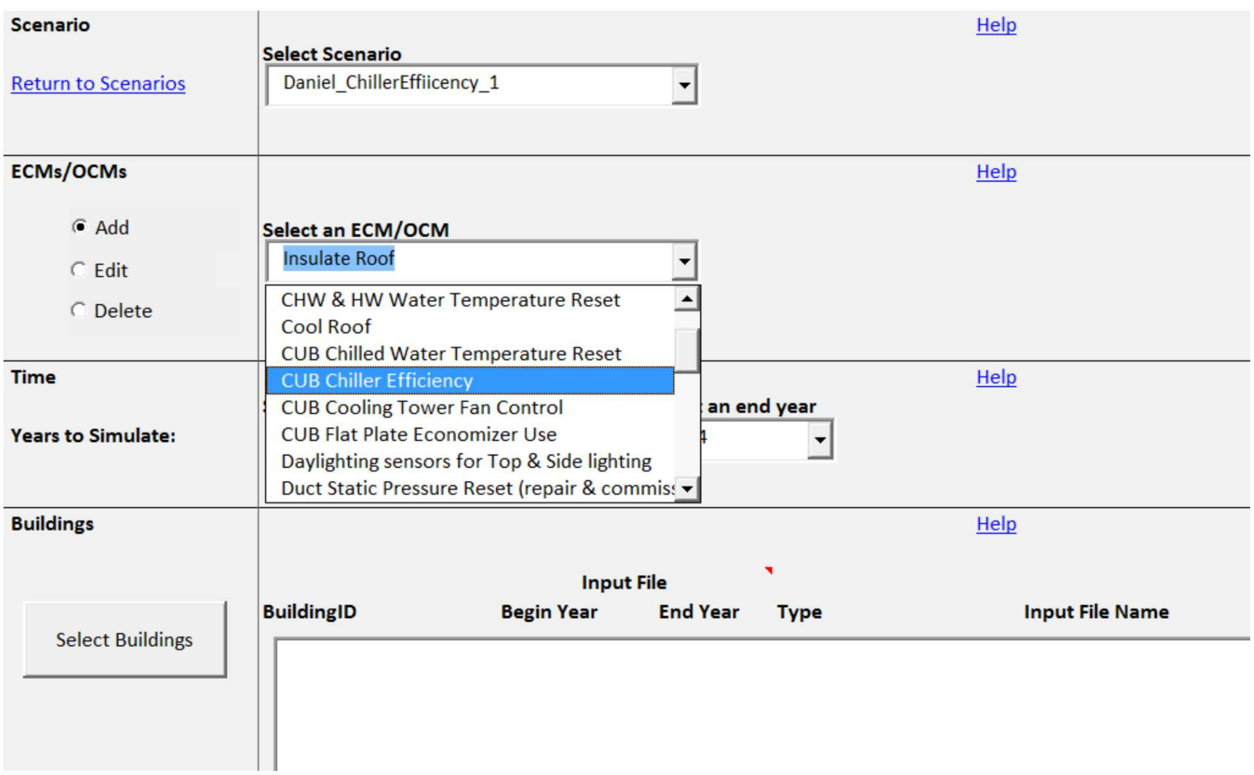


Figure 52. Selecting an ECM

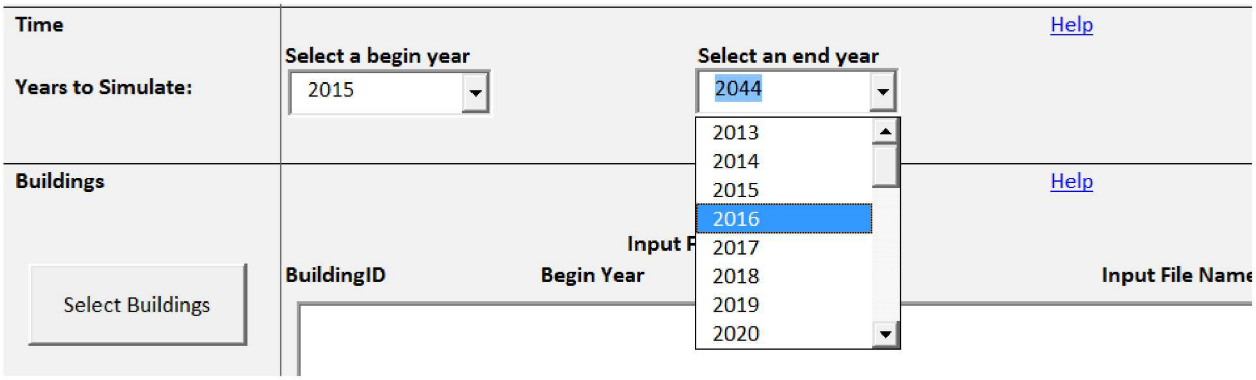


Figure 53. Select begin and end year

Press the "Select Buildings" button. A dialogue box opens allowing several functionalities to select a set of buildings as seen in Figure 54. In this demo, we will select buildings 962 and 981 as seen in Figure 55. Press "Done" when the desired buildings have been selected. The list box next to the "Select Buildings" provides information about the buildings that have been selected. A combo box to the far right top of this list box can be used to display additional attributes of the buildings including calibration status as seen in Figure 56.

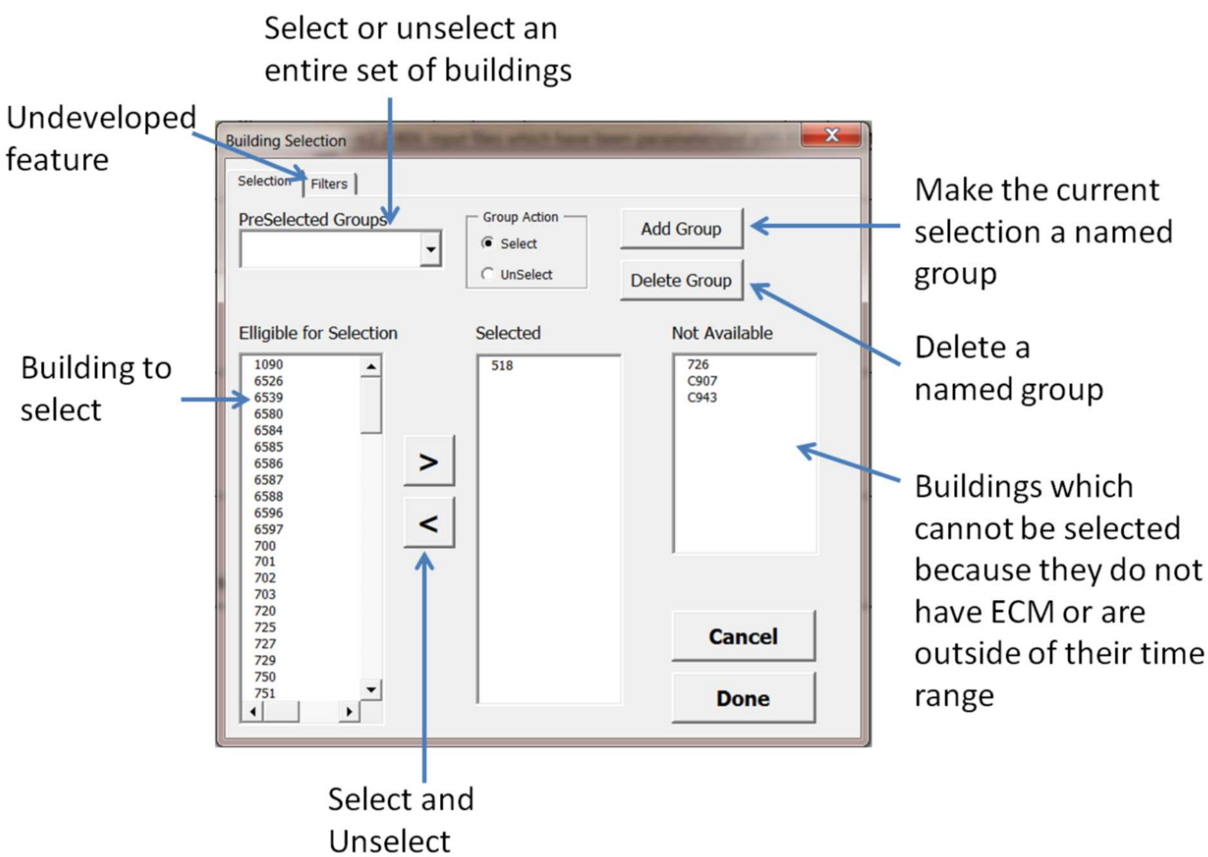


Figure 54. Building selection form features

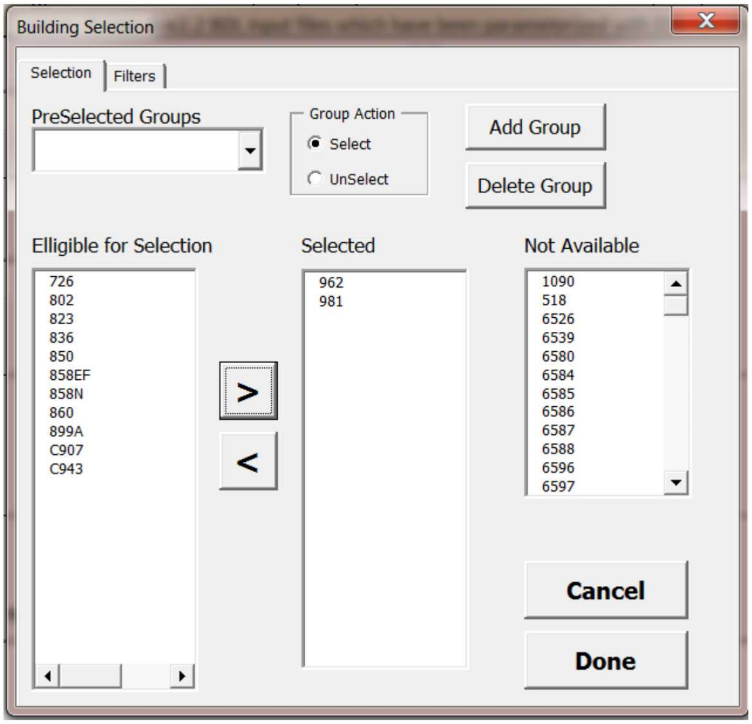


Figure 55. Buildings selected

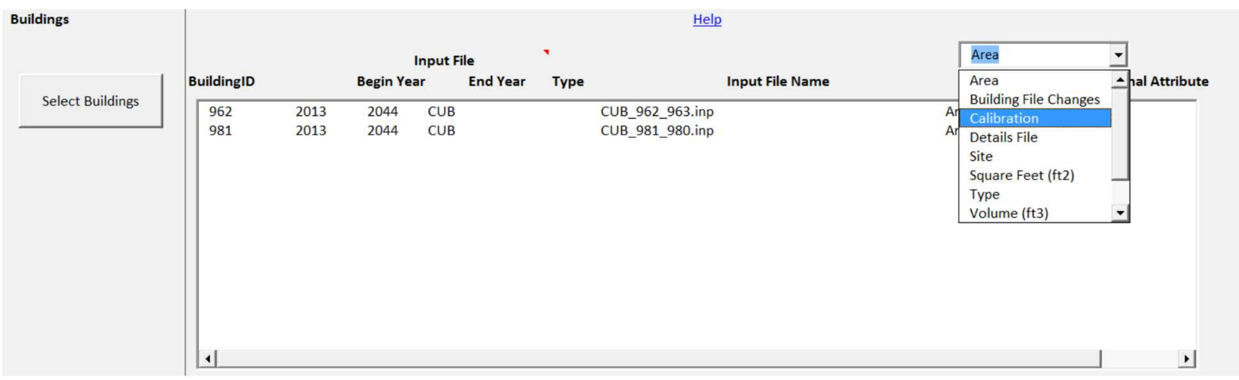


Figure 56. Building information listbox

Now that the ECM, years, and buildings have been selected, we are ready to scroll down to the second half of the Input sheet. Data input is accomplished through a dynamic table whose entries are refreshed from tables in the database (i.e., the values input are only temporary in Excel). First select a parameter to change as seen in Figure 57. It is necessary to understand what each ECM parameter means. Some of the ECMs have explanations provided in this user manual. Enter data for each year as desired. Changes will show up in red as seen in Figure 58. You can also view parameters listed as “read only.” These are the actual parameters that go to the building input files. There is a relationship between these “read only” parameters and the input parameters that are sometimes a simple one to one relationship but other times may be controlled by a functional relationship such as an equation or even a VBA routine. This is intended to make the input parameters easier to understand.

Change Chiller 1 Efficiency to 0.5 for both buildings. Once the required changes have been made, press, “Commit Changes,” to send the current table to the database.

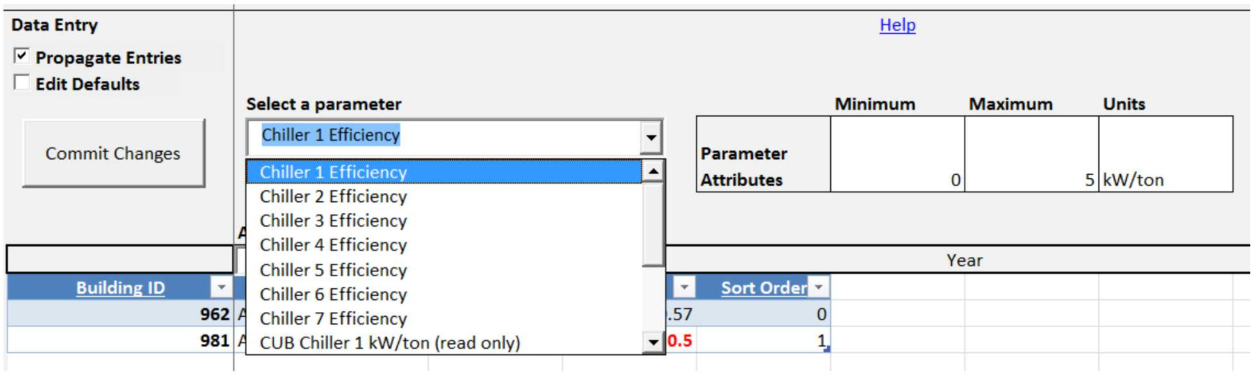


Figure 57. Selecting an ECM Parameter

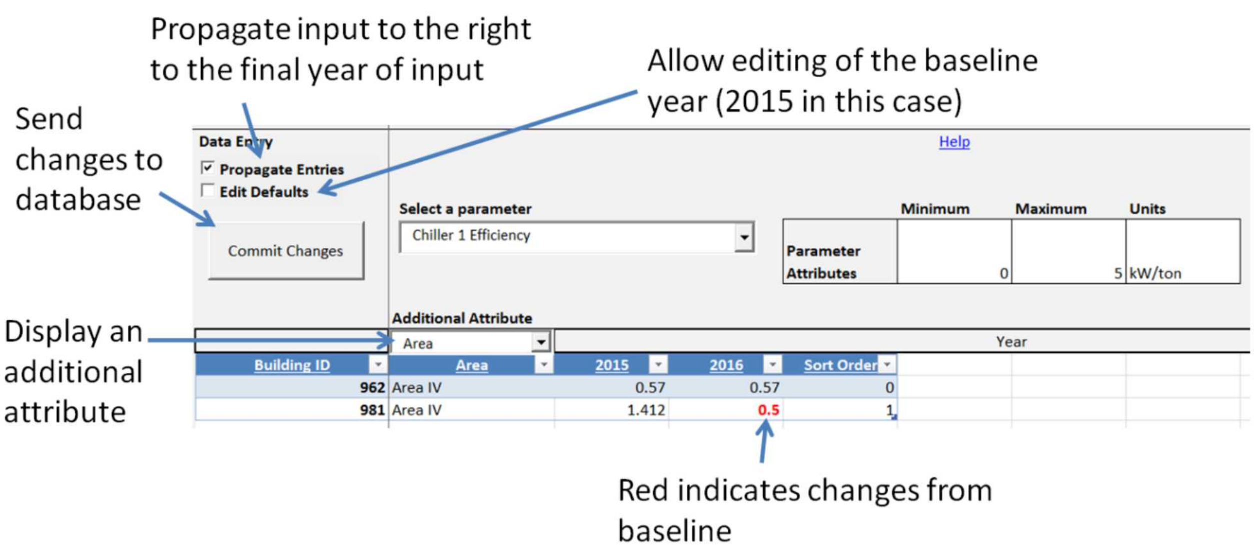


Figure 58. Data Entry Features

Once all parameters have been changed, you can return to the top of the page and “Add” another ECM if more than one ECM is desired. For this tutorial we move on to the Run sheet. A simple click on the “Run” tab or returning to the scenario page and right clicking on the scenario and selecting “Run” can accomplish this, as seen in Figure 59.

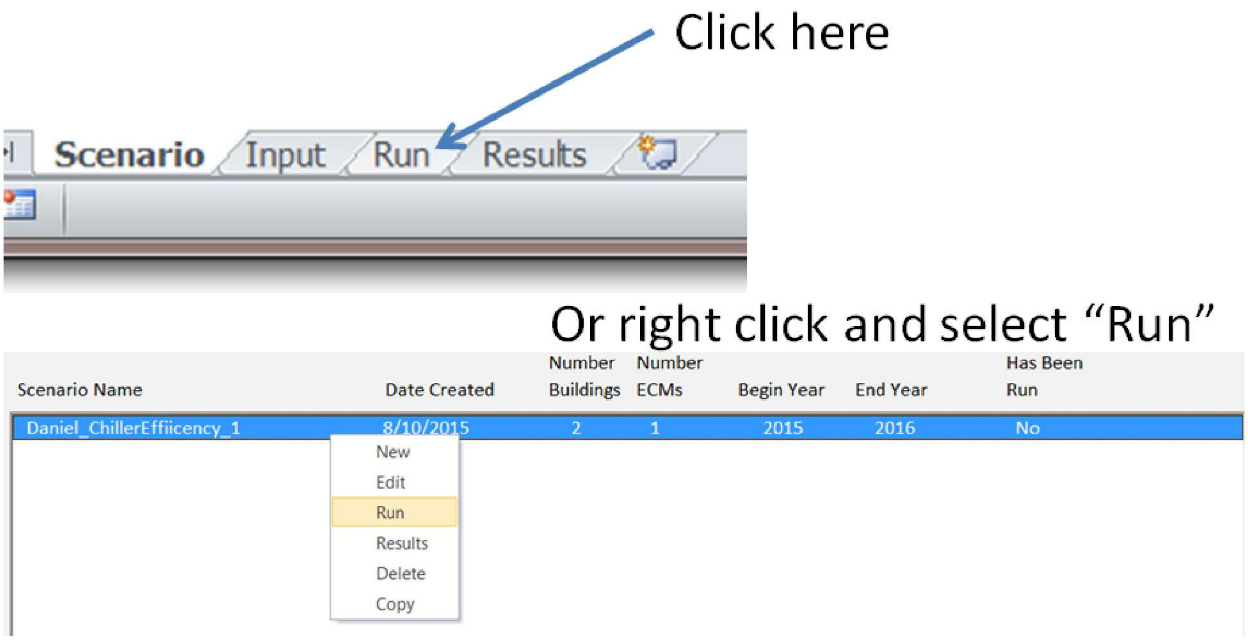


Figure 59. Getting to the run sheet

The run sheet has three functions: review output, select output, and run IX as seen in Figure 60. Click on "Review Input" to view all of the changes made for the scenario in a single pivot table as seen in Figure 61. The sheet creates a pivot table of all of the changes made to parameters. This sheet is very useful to look at all of the input for scenario in a single table. Often a single look at this table will reveal an incongruity between what you think your input is and what the actual input is. No changes to the input can be made here. Click on "Return to Run Page" in the upper left hand corner. Next, click on the "Select Output" button. A form opens up as seen in Figure 62 that allows the selection of various end uses for gas and electricity consumption. For now, just leave total electricity and total gas selected and press the "Continue" button. Finally press the "Run IX Scenario" button. A progress meter as seen in Figure 63 should pop up and the scenario should run to completion.

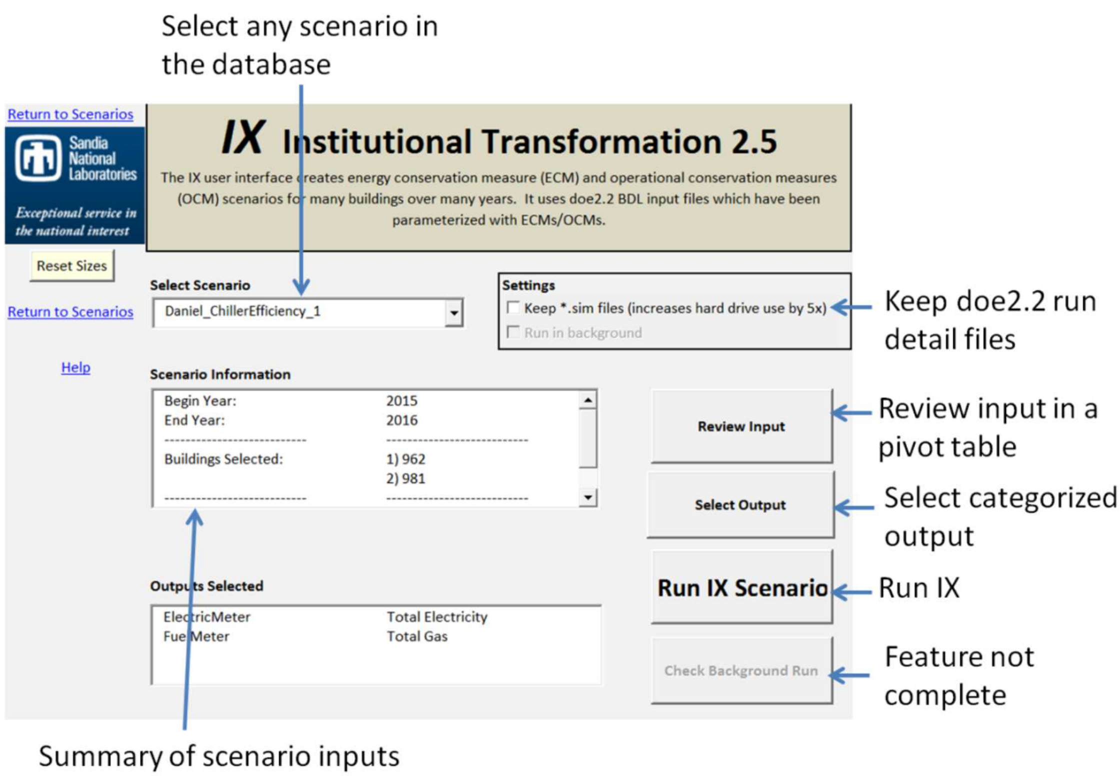


Figure 60. Run sheet features

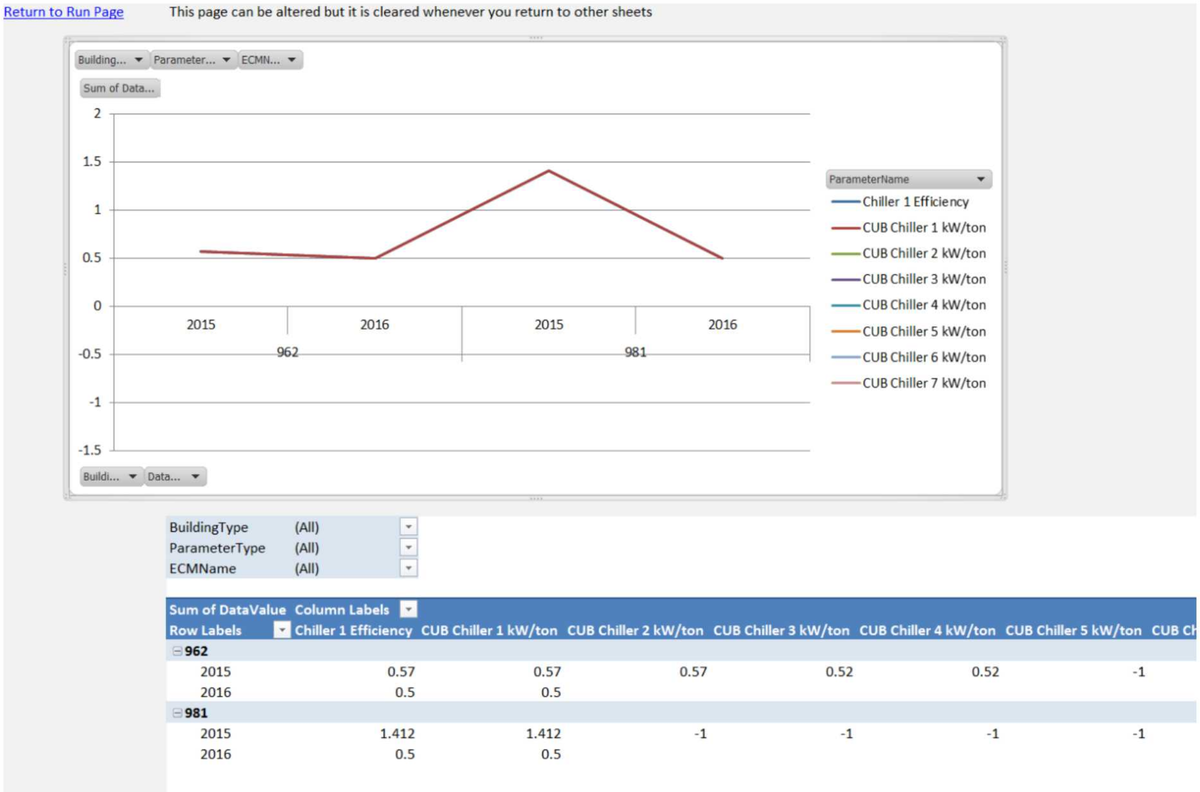


Figure 61. Input pivot table

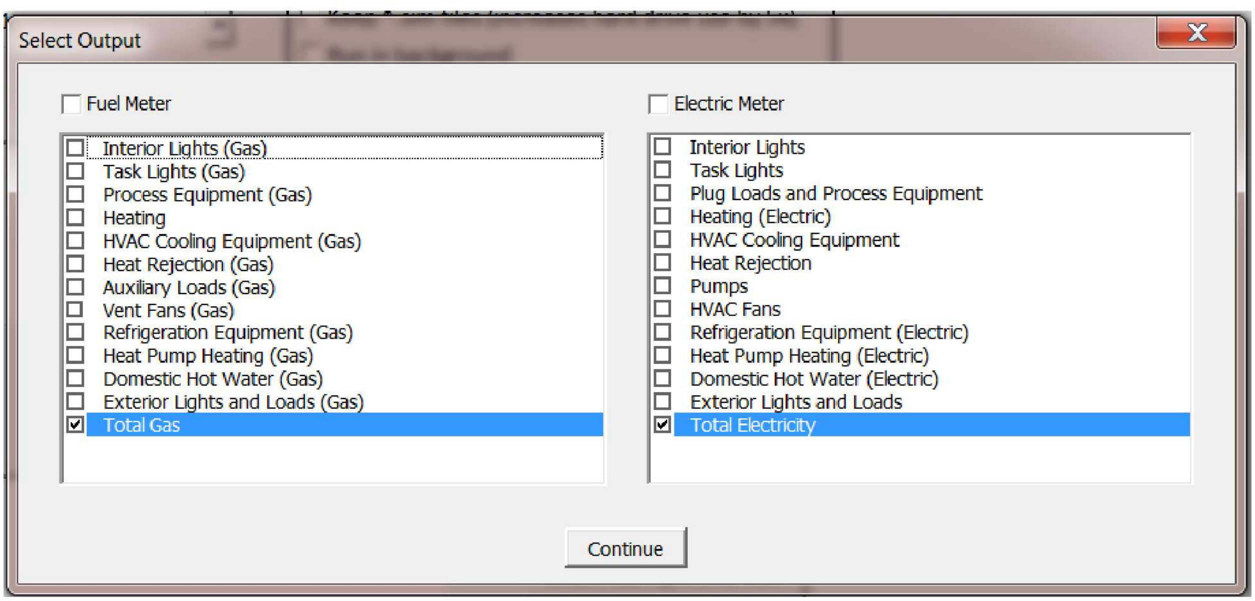


Figure 62. Select output form

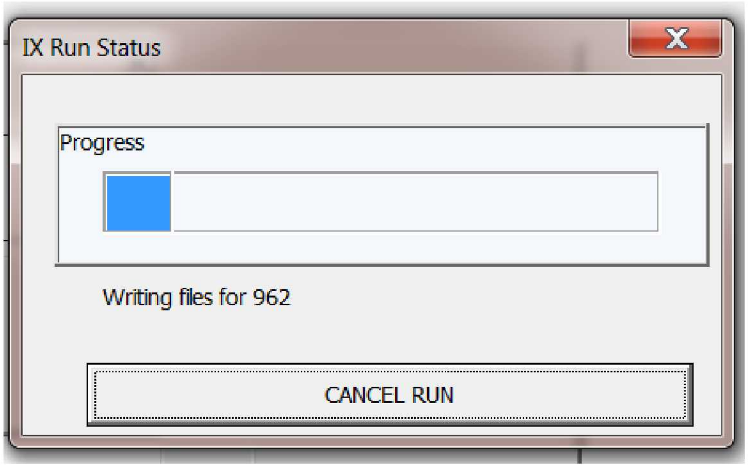


Figure 63. Progress meter

We can now enter the “Results” sheet either by returning to the Scenario sheet and right clicking and selecting “Results” or by simply picking the “Results” sheet tab. A view of the results sheet is provided in Figure 64. The results are presented in a pivot table and pivot chart format. Pivot tables enable quick rearrangement of datasets which have known attributes. Each data entity can be assigned one of four attributes: report, legend, axis, and value. Report attributes allow summation over subsets of a given attribute. For example, if building ID is placed in the report box, then the pivot table will sum over every building that is selected for anything in the value box. Legend fields are displayed as columns in the pivot table and are displayed in the legend in the pivot chart. Axis fields display on the axis of the pivot chart and form rows. There are many tutorials on how to use pivot tables online.

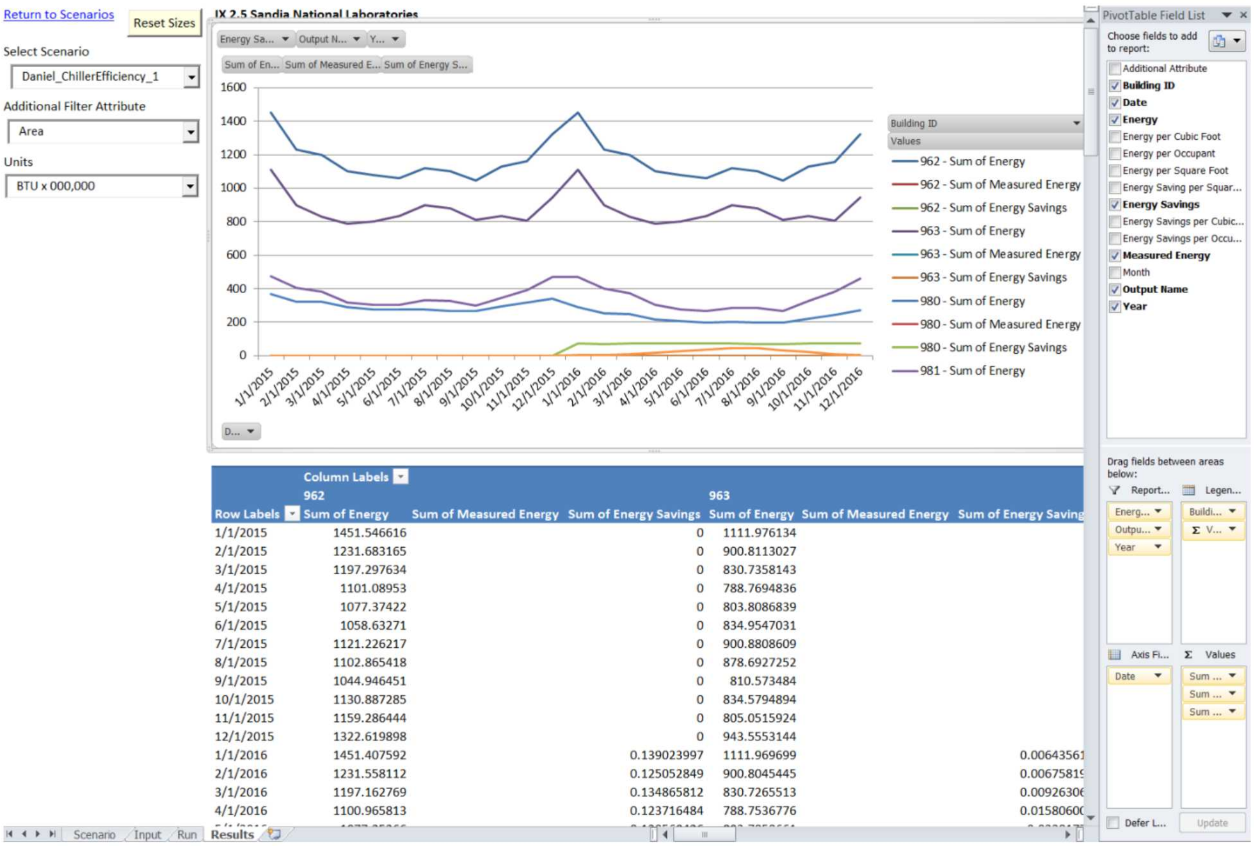


Figure 64 Results sheet for scenario run

Distribution

1 Marlin Addison
Arizona State University
PO Box 872102
Tempe, AZ 85287-2103

1 Bridgers and Paxton Consulting Engineers, Inc.
4600 Montgomery Blvd NE
Albuquerque, NM 87109

1	MS0730	Douglas W. Vetter	04144 (electronic copy)
1	MS0924	Gerald R. Gallegos	04853 (electronic copy)
1	MS0924	Jack Mizner	04853 (electronic copy)
1	MS0928	Christopher A. Evans	04844 (electronic copy)
1	MS1137	Will Peplinski	06926 (electronic copy)
1	MS1138	Daniel Villa	06926 (electronic copy)
1	MS0899	Technical Library	09536 (electronic copy)

Sandia National Laboratories