# The BEE Software Collaborative: An Open Source, Rule-Based Architecture for Building Energy Efficiency

*Martha Brook, California Energy Commission*
*Scott Criswell, Wrightsoft Corporation*

**ABSTRACT**

In the Building Energy Efficiency (BEE) software domain, there is an emerging collaboration supported by government, utilities and key industry organizations with developers from both the public and private sector. This collaborative is producing and organizing building component and climate data, along with energy simulation tools and a core software library. This library includes rule sets and rule processing software that operate on a building data model to spawn energy simulations that conform to the rules. Some rule sets contain simple default assumptions for most building parameters. Others implement design constraints through logical operations on data model elements. The first products from this collaborative will be public and private software tools that implement building energy efficiency code compliance in California. This paper introduces the data, models, rules and software that comprise this shared open source library. An invitation to join the collaborative is extended by illustrating how the BEE software can support a variety of programs and products in the energy efficiency modeling domain.

## Introduction

Performance-driven building energy design, whether a goal or a mandate, requires robust modeling tools. Both single building energy design studies and sector level energy policy analyses can be more successful if these tools facilitate multiple building energy simulations, comparisons to benchmarks and design targets, and the application of constraints and intelligent defaults. This success is not often realized because these building modeling tools, if they exist at all, are proprietary, such that only building design projects with substantial budgets can afford their use. Resource constraints and competitive bidding requirements also keep governments and other policy makers from accessing and updating these tools for policy analysis and program implementation.

California's long history of performance-based energy efficiency standards has resulted in a building design community that expects to have flexibility in meeting these standards. This flexibility can enable design innovation, but only if the modeling tools used for code compliance can simulate the energy performance of innovative designs and apply the required code constraints. There should also be regular feedback during the iterative design process as to whether and to what extent design alternatives meet or exceed code. Although California has thirty years of experience with performance standards, its building designs have not benefited from these seemingly requisite modeling processes. The building energy models used in code compliance software typically lag behind technology advances, and the process of proving that buildings meet code is separate from, rather than integrated with, typical design regimens.

California energy policy includes aggressive goals to design and construct Zero Net Energy buildings, by 2020 for residential buildings and by 2030 for nonresidential buildings (CEC, 2012). Beyond California, across the nation and beyond, climate policies and environmental stewardship are driving the need for high performance buildings. Developing policy requirements for, and designing high performance buildings will require better collaboration between government and industry, to leverage limited resources.

All market actors involved in the design of and mandate for high performance buildings can benefit from the ability to produce multiple simulations, apply design assumptions and policy requirements to building models, and compare results to design targets. We call the aggregate of these model-based processes "rule-based analysis," a term meant to encompass both building energy simulation and structured modifications to the building models simulated. Modeling tools that implement rule-based analysis for design and product performance investigations can be used by a diverse set of market participants. Architects, designers, product manufacturers, energy consultants, code developers, and program implementers motivated to improve the energy performance of buildings all need access to, and the ability to modify, tools that facilitate rule-based analysis.

## Background

### Performance Standards in California

The Warren-Alquist Act requires the California Energy Commission (CEC) to "develop a public domain computer program which will enable contractors, builders, architects, engineers and government officials to estimate the energy consumed by residential and nonresidential buildings" in order to implement California's Building Energy Efficiency Standards (Standards, aka Title-24) (CEC, 2007). The Standards include a performance-based compliance option that requires the use of software that is certified by the CEC for this purpose.

To meet the WAA mandate, the CEC must (1) establish reference methods for modeling the energy-related features of building designs, and (2) provide publicly available Standards compliance software. The reference methods establish the engineering basis for estimating the hourly energy use of residential and nonresidential buildings across California's climate regions and are used to determine the energy cost savings expected from updates to the Standards. The CEC also uses these reference methods (separate methods for residential and nonresidential buildings) as the basis of comparison during the compliance software certification process, where private vendors can submit their own software for consideration as additional compliance tools.

For the 2013 Residential Standards reference method, California's investor-owned utilities collaborated with the CEC to co-fund the development of the California Simulation Engine (CSE). CSE is a first principles residential building simulation tool without the simplifying assumptions for solar gains and mass transfer of the previous residential reference method, CALRES. CSE is a set of building energy simulation algorithms currently available in open source software (Wilcox, 2010).

The 2013 Nonresidential Standards reference method is EnergyPlus™, the building simulation tool supported by U.S. DOE. The CEC's previous reference method was DOE2.1e, a

tool no longer publicly supported by either U.S. DOE or the CEC. EnergyPlus as the reference method will allow the CEC to establish a basis of comparison for modeling a much broader set of building energy technologies and control systems than was possible in previous code cycles. EnergyPlus is also available under an open source software license (LBL, 2012).

California's performance-based compliance process requires a proposed building design to be compared to a "standard" building design, which is the proposed design modified to just meet the mandatory and prescriptive requirements of the Standards. The compliance software must compute annual energy budgets for each design and produce comparative results. If the annual energy budget of the proposed design is equal or less than that of the standard design, the candidate building design complies with the Standards. The implementation of California's performance standard is accomplished with rule-based analysis software. Historically, the CEC has met its mandate to provide Standards compliance software to the public by obtaining a limited license to proprietary software from vendors who also provide private Standards compliance software to the market.

**Other Rule-based Software for Code Compliance**

Rule-based analysis software has been used over the last fifteen years to implement multiple performance standards in the U.S. and Canada. A few of these efforts resulted in proprietary software tools with the common feature of being able to, outside of compiled source code, edit and deploy a set of rules that modify a building model for a proposed design, and generate a separate standard design building model. COMcheck-Plus, developed by Regional Economic Research for the Pacific Northwest National Laboratory, implemented the ASHRAE 90.1-1989 Energy Cost budget in 1997-99. eQUEST$_{tm}$, a graphical user interface to the DOE-2.2 simulation engine developed by Jeff Hirsch and Associates, adapted the COMcheck-Plus rule-based analysis features to develop a California Title-24 energy code analysis ruleset that was certified for Standards adopted in 2001, 2005 and 2008. eQUEST is distributed as freeware, but is not currently an open-source product (http://doe2.com/equest/). Natural Resources Canada is now in the process of developing an eQUEST derivative, CAN-QUEST, which will test compliance to Canada's 1997 MNECB and NECB 2011 energy codes.

**Open Source Software Collaboratives**

The majority of application software used for building energy modeling by end-users is proprietary. While this usually includes the benefit of software support, there are significant disadvantages when public agencies have mandates to develop, maintain and update software for program implementation. Proprietary software solutions typically result in requiring ongoing sole source contracts with limited source code and distribution licenses. When several public agencies or their delegates need to modify each other's software tools for specific implementations, they are forced to enter into additional sole source agreements that also have limited license to modify and distribute the proprietary software. Further, if agencies do not own the software tools they fund, the public cannot access and make use of the source code to produce derivative works, thereby limiting the value that publicly funded software projects provide.

Open source software projects are becoming popular in many sectors. This quote from NASA's Open Government Initiative describes the benefits that can arise from public and private collaboration, which is not specific to NASA but applicable to open source software collaboratives in general: *"Open source development-which allows free access to software source code to allow anyone to make improvements-is revolutionizing the way software is created, improved, and used. The open source software movement is inherently transparent, participatory, and collaborative. Open source at NASA gives the public direct and ongoing access to NASA technology... [NASA should]shift our open source activities from its one-way direction of giving the public access to finalized software products, to allowing two-way collaboration as part of the development process. The benefits of allowing the public to assist in development of NASA software include increased software quality, accelerated software development, and a higher rate of technology transfer both to and from NASA."* (NASA, 2012)

OpenStudio is a great example of a government funded project in the building energy design domain that provides access to source code and executable applications to facilitate private sector use. OpenStudio is a cross-platform collection of software tools that facilitates building energy modeling using EnergyPlus and Radiance (NREL, 2012). OpenStudio is not currently itself a collaborative, in that multiple organizations or individuals outside of NREL staff do not contribute to the code base, but OpenStudio is available under a Lesser General Public License and derivative works are being developed from this software.

**The Building Energy Efficiency (BEE) Software Collaborative**

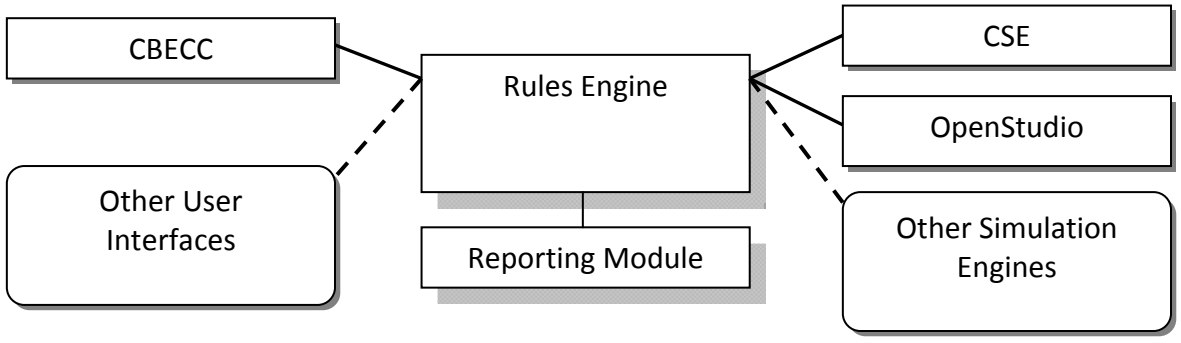
The CEC has made a commitment to collaboratively develop, test, document, and support open source building energy modeling software and other building energy analysis tools used for Standards development, Standards compliance and other energy efficiency public policy implementation. The BEE Software collaborative members are other funding partners managing public goods energy efficiency programs, software vendors interested in adding code compliance functionality into their design tools, and building scientists, data modelers and software developers constructing the open source software architecture.

For the 2013 Standards, the CEC will use the BEE Software to construct the compliance software for both the residential and nonresidential Standards. This California Building Energy Code Compliance (CBECC) software will use CSE for residential and EnergyPlus (via OpenStudio) for nonresidential building energy modeling, then add a software layer that implements the performance compliance rules. The BEE Software architecture separates rules processing from the energy simulation, so it will support modifications of the rules to implement other programs sponsored by collaborative members. This architecture also separates the rule processing software from both the simulation tool and the user interface via application programming interfaces (APIs), so collaborative members can employ alternate user interfaces to access the rule-based analysis capabilities, and also employ the rules processing software with different simulation tools.

## BEE Software Architecture

The BEE Software comprises four principal components: User Interface, Rules Engine, Simulation Engine, and Report Module. Figure 1 illustrates at the highest level how these modules combine for the BEE Software. The solid lines in Figure 1 indicate the current software implementation with CBECC, and the dashed lines indicate where collaborative members are expected to create new BEE Software implementations.

**Figure 1. High Level BEE Software Architecture**




**The User Interface (UI)** is simple, relative to the detailed building and system description processes facilitated by several building energy design tools. The BEE Software is not intended as a design tool, although its functions can readily support parametric analyses and apply design regimen rules. The collaborative, to date, is providing a "bare bones" interface that may of course be expanded in the future. The user will have the ability to describe building features with the UI as necessary to populate the Rules Engine data model. Users will be able to read/open/edit Rules Engine building models written from other building design, simulation, and analysis tools if these tools can export the Rules Engine data model via XML. This should facilitate importing complex building models, a welcome alternative to creating them from scratch within the UI. OpenStudio, for example, exports the nonresidential Rules Engine building model. The UI Tool is designed for the Windows$^{tm}$ operating system as a desktop/client program.

**The Rules Engine** is the key component that allows rules to be applied to building models. This software module can be integrated into other tools and interfaces to perform a variety of rule-based Building Energy Modeling (BEM) analyses. The Rules Engine uses a building data model that is simulation engine agnostic, to facilitate its use within multiple design tools. This is an important element of the BEE Software – to meet the objectives of the collaborative, rule generation cannot require knowledge of the specific terms used at the simulation engine level. Another important feature of the Rules Engine is that it reads in the rulesets at run time; the rulesets are not included in the compiled software code. Multiple rulesets can be used within a single instance of the executable software, as long as the data terms in the rulesets are consistent with the data models supported by the Rules Engine (the Rules Engine currently supports separate data models for residential and nonresidential buildings). This functionality provides a great amount of flexibility and ease of use for the collaborative.

Members can experiment with different performance rules in multiple rulesets without editing or recompiling source code.

**The Simulation Engine** is more than just the BEM tool that produces hourly simulations of building energy use. This component performs the following functions: (a) translates the Rules Engine data model into the simulation engine input data language, (b) manages the simulations, and (c) retrieves the simulation results, maps them back into the Rules Engine data model, then returns them. The CBECC software will connect with CSE for residential energy simulations and EnergyPlus, via OpenStudio, for nonresidential energy simulations.

**The Reporting Module** will accept data model elements and analysis results from the Rules Engine, then use that data to populate standard reports. The Reporting Module will write these reports to one or more common formats (such as PDF, or XML/XSD/XSL files) which will then be made available through the User Interface for display, print, or transmission to another application. The Reporting Module is the least developed of all the BEE Software components, to date. Currently, the functional requirements for the Reporting Module meet the specific needs of California's performance code compliance software. These functions can be expanded to meet other data reporting and results visualization needs of collaborative members, given time and resources. The benefit of this software architecture, where the Rules Engine calls the Reporting Module, is that third party software applications that incorporate the Rules Engine to add code compliance (or other rule-based design processes) will not need separate reporting capabilities.

## Key Components of the BEE Software

### Rules Engine

At the heart of BEE Software is the Rules Engine, which performs the following functions:

- Confirms building model validity before rules processing begins,
- Establishes rule evaluation order from rule and attribute dependencies,
- Processes rules to generate building models ready for simulation,
- Manages building model transformations required to implement the rules,
- Manages simulations,
- Reports messages and results to the calling application.

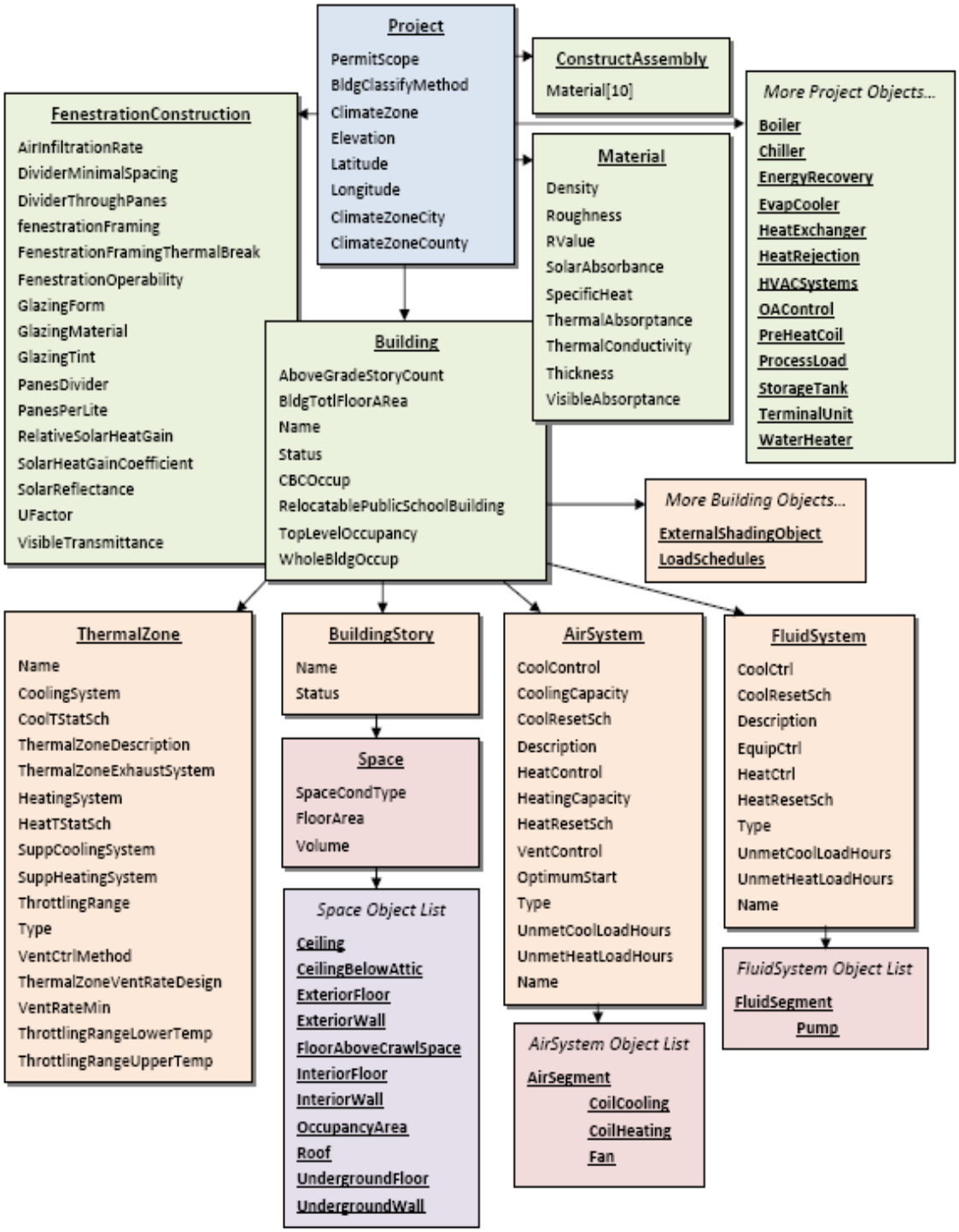
The Rules Engine API enables tools to send and receive data models. The API also allows the User Interface to retrieve needed display data from the ruleset, such as building component descriptions, valid numeric ranges, and units.

### Building Data Models

The data models used to describe building components in the BEE Software contain only as much detail as is necessary to assign rules to energy-related attributes. The data models are not intended to include the detail necessary to be used directly in energy simulation, although

they are made up of a fairly broad and complex range of information, as illustrated by the summary of the nonresidential building model in Figure 2.

**Figure 2. Summary of the Nonresidential Building Model Objects, Properties and Relationships**



The nonresidential data model was developed by reviewing relevant data exchange and data model standards (gbXML and IFC), the COMNET Modeling Guidelines and Procedures, OpenStudio's internal data model, and Title 24, Part 6 (gbXML, 2012; IFC, 2012; COMNET, 2010 ; NREL, 2012; CEC, 2008). The collaborative believes that incorporating existing data model components, where appropriate, and establishing an explicit map between the BEE software data model and these existing data models, when a direct incorporation is not possible, will allow the BEE software products to be readily integrated into other industry tools.

The residential data model development relied on the collaborative members' extensive experience modeling residential energy consumption for Standards development and compliance. While there are nascent data exchange standards for the residential building energy domain (e.g. HomePerformanceXML), these do not have the required hierarchical structure or detailed attributes needed for rule-based energy analysis (BPI, 2010). The BEE Software includes separate data models for residential and nonresidential buildings because (1) the rule assignments expected to be needed by collaborative members for each require a different set of building energy attributes, and (2) the data models native to CSE and EnergyPlus that the BEE Software data models must eventually be translated into are significantly different.

**Implementing Multiple Data Models and Rulesets**

The BEE software can support multiple data models and rulesets without requiring the modification (and recompilation) of the Rules Engine and User Interface source code. The fact that text files are used to define both the scope of the rule-based analysis and the corresponding user entries allows the software implementations to be managed by domain experts rather than software developers.

Translations of the Rules Engine data model to and retrieval of results from simulation tools are completed in modules tightly integrated with the Rules Engine. The residential building model translation is done within the ruleset data model itself, which includes parallel object definitions, one set used to interface with users (via CBECC and third party user interfaces) and a second set that map directly to CSE inputs. The nonresidential building model is exported from the Rules Engine in the form of an XML file and translated for EnergyPlus within OpenStudio. Collaborative members who wish to combine rule-based analysis with their own simulation engines need to create translations of the Rules Engine data model to the equivalent building component descriptions native to their simulation tools.

**Ruleset Structure**

A ruleset is represented by a series of files that contain energy code data and logic in the form of rule expressions and look-up tables. The source versions of these files are text (.txt & .csv) which make them very easy to view, edit, compare and track in version control systems. The ruleset source files are combined into encrypted binary files when distributed with software that use or integrate the Rules Engine, in order to ensure the integrity of the compliance analysis. Some of the data that is contained in a compliance ruleset is as follows:

- **Ruleset ID & version.** Identifiers available at runtime to identify the ruleset and its version;
- **Look-up Tables.** A series of text/CSV files containing a wide variety of data defined by the rule authority (e.g. the energy code) that can be referenced by rules in the ruleset;
- **Range Checks.** The definition of range limits (message, warnings and errors) and conditions in which those limits are to be applied to building model component attributes;
- **Component Libraries.** Individual building component definitions (consistent with rule requirements) that can be imported into building models generated by the ruleset;
- **Rules.** Lists of rules that control how a building model is manipulated (e.g. to determine its compliance with the energy code).

**Rule Expressions**

The following tables provide a summary of the expression syntax and functionality of the rules applied to building model attributes that are processed by the Rules Engine.

**Table 1. Arithmetic and Logical Expression Operators**

Functional Requirements for ACM Standards Compliance Engine Software, Version 0.6 (Criswell et al., 2011).

| | | | |
|---|---|---|---|
| ***Arithmetic:*** | * | Multiplication | |
| | / | | Division |
| | + | | Addition |
| | - | Subtraction (or Unary Minus) | |
| | % | Remainder (mod) | |
| | ** | Exponential | |
| ***Logical:*** | \|\| or .OR. | Or | |
| | && or .AND. | | And |
| | ! or .NOT. | Not | |
| | == or .EQ. | Equal | |
| | != or .NE. | Not equal | |
| | >or .GT. | Greater than | |
| | <or .LT. | Less than | |
| | >= or .GE. | Greater than or equal to | |
| | <= or .LE. | Less than or equal to | |

**Table 2. Standard Expression Functions**

Functional Requirements for ACM Standards Compliance Engine Software, Version 0.6 (Criswell et al., 2011).

| | | |
|---|---|---|
| | abs( x ) | Absolute value |
| | max( x1, x2 ) | Maximum |
| | min( x1, x2 ) | Minimum |
| | int( x ) | Rounds x to nearest integer |
| | ftoa( x ) | Converts a floating point number to a character string |
| | strlen( x ) | Number of characters contained in string x |

| | | |
|---|---|---|
| | strlower( x ) | Returns an all lower case version of string x |
| | strupper( x ) | Returns an all upper case version of string x |
| | log( x ) | Natural logarithm |
| | log10( x ) | Base-10 logarithm |
| | exp( x ) | Exponential (e raised to the power x) |
| | pow( x1, x2 ) | Power (x1 raised to the power x2) |
| | mod( x1, x2 ) | Modulus (remainder) of x1 / x2 |
| | sqrt( x ) | Square root |
| | sin( x ) | Sine (angle expressed in radians) |
| | asin( x ) | Arcsine (result in radians) |
| | cos( x ) | Cosine (angle expressed in radians) |
| | acos( x ) | Arccosine (result in radians) |
| | tan( x ) | Tangent (angle expressed in radians) |
| | atan( x ) | Arctangent (result in radians) |
| | *Note:* All function names are case insensitive. | |

**Table 3. Executable Expression Statements**

Functional Requirements for ACM Standards Compliance Engine Software, Version 0.6 (Criswell et al., 2011).

**if** (*expression*) **then**
*statement*
**else** **if** (*expression*) **then**
*statement*
**else**
*statement*
**endif**
**endif**
*Note:* Each if statement must contain an else.

**switch** (*expression*)
**case** (*const*) : *statement*
**case** (*const*) : *statement*
**default** : *statement*
**endswitch**
*Note:* Each switch statement must contain a default.
*Note:* The individual case values listed as *const* must be either numeric constants or enumerations which can be directly translated into numeric constants at parse-time in order to prevent excessive evaluation-time error checking and minimize the complexity of the statement.

*Notes*: All reserved words in bold (if, else, endif, ...) are case insensitive.
If and Switch statements can be nested as long as the expression evaluates to a single return value.

## Conclusion

The BEE software infrastructure can support a myriad of public and private efficiency programs and products. Data models, building component properties, system designs, rule sets and user interfaces can be shared and expanded to meet product specific needs. The Wrightsoft Corporation is in the process of leveraging the BEE software for a product designed to test compliance of single and multi-family homes to the residential energy code in the state of Florida. This product will take advantage of CSE to perform hourly simulations of energy use and will include a customized ruleset consistent with Florida's IECC-based energy code. This is the first example of a collaborative member taking advantage of this open source software to develop a new building energy efficiency software product.

The BEE software can also support other rule-based analysis other than energy code compliance. A few obvious candidate applications that can leverage the BEE software are beyond-code programs such as LEED, or California's New Homes and Savings By Design programs. Building asset rating systems can also be readily implemented using this rule-based software architecture. Also, private design firms can establish their own set of rules of default assumptions and performance benchmarks for application in their high performance building design processes. Technology manufacturers can also access and modify code rulesets to test and propose how new products should be modeled in an energy code context, which will facilitate these technologies' incorporation into future performance standards.

## References


[BPI] Building Performance Institute. 2010. Schema documentation for HomePerformance_XML.xsd.http://www.homeperformancexml.org/sites/default/files/HomePerformance%20_XML-2%20Schema%20Documentation_0.pdf.

[CEC] California Energy Commission. 2007. Warren-Alquist State Energy Resources Conservation and Development Act. Sacramento, CA: CEC-140-2007-004.

[CEC] California Energy Commission. 2008. 2008 Energy Efficiency Standards for Residential and Nonresidential Buildings. Sacramento, CA: CEC-400-2008-001-CMF.

[CEC] California Energy Commission. 2012. 2011 Integrated Energy Policy Report. Sacramento, CA: CEC-100-2011-001-CMF.

[COMNET] Commercial Energy Services Network, 2012. Commercial Buildings Energy Modeling Guidelines and Procedures. RESNET Publication 2010-001.

Criswell, Scott A. et al. 2011. Functional Requirements for ACM Standards Compliance Engine Software, Version 0.6. San Francisco, CA: Architectural Energy Corporation.

[gbXML] The Open GreenBuilding XML Schema, Inc. 2012. http://www.gbxml.org/currentschema.php.

[IFC] buildingSMART International Ltd. 2012. Industry Foundation Classes. http://buildingsmart.com/standards/ifc.

[LBL] Lawrence Berkeley National Laboratory. 2012. EnergyPlus™ Open Source License v1.0. Berkeley, CA.

[NASA] National Aeronautics and Space Administration.2012. Open Government Initiative. http://www.nasa.gov/open/plan/.

[NREL] National renewable Energy Laboratory. 2012. OpenStudio. http://openstudio.nrel.gov/.

Wilcox, Bruce A. 2011. 2013 Residential Standards Development Software. http://www.energydataweb.com/consortium/Documents/Wilcox110114SDP.pdf.