ENERGYPLUS™

Getting Started with EnergyPlus

Everything You Need to Know about Running EnergyPlus

(and a start at building simulation)

Date: April 8, 2003

COPYRIGHT © 1998-2002 THE BOARD OF TRUSTEES OF THE UNIVERSITY OF ILLINOIS AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA THROUGH THE ERNEST ORLANDO LAWRENCE BERKELEY NATIONAL LABORATORY.

ALL RIGHTS RESERVED. NO PART OF THIS MATERIAL MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS WITHOUT THE PRIOR WRITTEN PERMISSION OF THE UNIVERSITY OF ILLINOIS OR THE ERNEST ORLANDO LAWRENCE BERKELEY NATIONAL LABORATORY.

ENERGYPLUS IS A TRADEMARK OF THE US DEPARTMENT OF ENERGY.

TABLE OF CONTENTS

EnergyPlus Overview 1

What is EnergyPlus? 1

Why does EnergyPlus exist and what were its original goals? 2

Figure 1. EnergyPlus -- the big picture 4

Figure 2. EnergyPlus -- Internal elements 5

EnergyPlus Documentation Library 6

User Information Documents 6

Engineering Reference Document 6

Developer Information Documents 7

FAQ Document 7

Getting Started with EnergyPlus 8

Running EnergyPlus 9

Introduction 9

EP-Launch Program 9

Figure 3. EP-Launch Screen 10

Start EP-Launch 10

Selecting Input and Weather Files 10

Running a Simulation 10

Looking at the Results 10

Figure 4. EP-Launch Finish Status. 11

Viewing the Drawing File without Running a Simulation 11

Editing the Input Files 11

Selecting Viewers and Editors 12

File Menu 12

Edit Menu 12

View Menu 12

TABLE OF CONTENTS

Help Menu 12

Recently Used Files 12

Caveats 12

When things go wrong 13

Figure 5. EnergyPlus crash within EP-Launch 13

Bugs 13

RunEPlus batch file 14

Running EnergyPlus by Hand 14

Running EnergyPlus Using a procedure file 15

EnergyPlus Files 15

Input Files 15

Output Files 16

Table 1. Error Message Levels – Required Actions 16

Postprocessing Program/Files 18

Information necessary to Run EnergyPlus 19

Building Simulation 19

A Methodology for Using Energyplus 19

Step 1: Plan Ahead 19

Step 2: "Zone" the Building 19

Figure 6. Fort Monmouth Education Center 20

Table 2. Zoning the Building by System Type 21

Figure 7. Thermal Zones in the Education Center 21

Figure 8. Single Zone Model of the Ft. Monmouth Education Center. ... 22

Step 3: Prepare to Construct the Building Model 22

Table 3. Surface types and categorization 23

Figure 9. Simplifications Using Equivalent Surfaces 26

TABLE OF CONTENTS

Table 4. Building Elements 27

Figure 10. Schematic of One Zone Model with Surface Information..... 28

Table 5. Compilation of Surface Information for the One Zone Model 29

Step 4: Compile Internal Space Gain Data 29

Table 6. Internal Gain Data 30

Creating Input Files.....31

IDD Conventions 31

Standard EnergyPlus Units 33

Table 7. Standard EnergyPlus Units 33

IDFEditor 34

Figure 11. IDF Editor Screen 35

Start IDF Editor 35

Creating or Selecting an Input Data File 35

Class List and Objects 35

Changing Values..... 36

Working with Objects 36

File Menu 36

Edit Menu 36

View Menu 36

Help Menu..... 37

Caveats 37

Bugs 37

BLASTTranslator..... 37

DOE2Translator 38

IFCtoIDF..... 39

Text Editor..... 39

TABLE OF CONTENTS

HVAC Templates 39

WeatherData 40

Table 8. Formats of weather data on the web site 40

Table 9. General Locations of Weather Sites on web page 40

PostProcessing 41

ReadVarsESO 41

PostProcessing Files details 41

eplusout.inp 41

eplusout.eso 41

eplusout.csv 42

DataFiles 43

Energy Meters 44

Figure 12. Illustration for Energy Meters 44

Table 10. Overall Meter Types 45

Table 11. Table of Metered Fuel Types 46

Table 12. End Use Category Types 46

Index 47

EnergyPlus Overview

Welcome to EnergyPlus! This section offers a big picture view of what EnergyPlus is, what it does, why it exists, and what its goals are AND provides an overview to the **EnergyPlus documentation library**. While this document does not provide every detail on the program, it does give the reader a macroscopic view of EnergyPlus and how it is intended to fit into the energy analysis and thermal load simulation software continuum.

What is EnergyPlus?

EnergyPlus has its roots in both the BLAST and DOE–2 programs. BLAST (Building Loads Analysis and System Thermodynamics) and DOE–2 were both developed and released in the late 1970s and early 1980s as energy and load simulation tools. Their intended audience is a design engineer or architect that wishes to size appropriate HVAC equipment, develop retrofit studies for life cycling cost analyses, optimize energy performance, etc. Born out of concerns driven by the energy crisis of the early 1970s and a recognition that building energy consumption is a major component of the American energy usage statistics, the two programs attempted to solve the same problem from two slightly different perspectives. Both programs had their merits and shortcomings, their supporters and detractors, and solid user bases both nationally and internationally.

Like its parent programs, EnergyPlus is an energy analysis and thermal load simulation program. Based on a user's description of a building from the perspective of the building's physical make-up, associated mechanical systems, etc., EnergyPlus will calculate the heating and cooling loads necessary to maintain thermal control setpoints, conditions throughout an secondary HVAC system and coil loads, and the energy consumption of primary plant equipment as well as many other simulation details that are necessary to verify that the simulation is performing as the actual building would. Many of the simulation characteristics have been inherited from the legacy programs of BLAST and DOE–2. Below is list of some of the features of the first release of EnergyPlus. While this list is not exhaustive, it is intended to give the reader and idea of the rigor and applicability of EnergyPlus to various simulation situations.

- **Integrated, simultaneous solution** where the building response and the primary and secondary systems are tightly coupled (iteration performed when necessary)
- **Sub-hourly, user-definable time steps** for the interaction between the thermal zones and the environment; variable time steps for interactions between the thermal zones and the HVAC systems (automatically varied to ensure solution stability)
- **ASCII text based weather, input, and output files** that include hourly or sub-hourly environmental conditions, IFC driven object definitions, and standard and user definable reports, respectively
- **Heat balance based solution** technique for building thermal loads that allows for simultaneous calculation of radiant and convective effects at both in the interior and exterior surface during each time step
- **Transient heat conduction** through building elements such as walls, roofs, floors, etc. using conduction transfer functions
- **Improved ground heat transfer modeling** through links to three-dimensional finite difference ground models and simplified analytical techniques
- **Combined heat and mass transfer** model that accounts for moisture adsorption/desorption either as a layer-by-layer integration into the conduction transfer functions or as an effective moisture penetration depth model (EMPD)
- **Thermal comfort models** based on activity, inside dry bulb, humidity, etc.

- **Anisotropic sky model** for improved calculation of diffuse solar on tilted surfaces
- **Advanced fenestration calculations** including controllable window blinds, electrochromic glazings, layer-by-layer heat balances that allow proper assignment of solar energy absorbed by window panes, and a performance library for numerous commercially available windows
- **Daylighting controls** including interior illuminance calculations, glare simulation and control, luminaire controls, and the effect of reduced artificial lighting on heating and cooling
- **Loop based configurable HVAC systems** (conventional and radiant) that allow users to model typical systems and slightly modified systems without recompiling the program source code
- **Atmospheric pollution calculations** that predict CO_2, SO_x, NO_x, CO, particulate matter, and hydrocarbon production for both on site and remote energy conversion
- **Links to other popular simulation environments** such as WINDOW5, COMIS (airflow model), TRNSYS, and SPARK to allow more detailed analysis of building components

More details on each of these features can be found in the various parts of the EnergyPlus documentation library.

No program is able to handle every simulation situation. However, it is the intent of EnergyPlus to handle as many building and HVAC design options either directly or indirectly through links to other programs in order to calculate thermal loads and/or energy consumption on for a design day or an extended period of time (up to, including, and beyond a year). While the first version of the program contains mainly features that are directly linked to the thermal aspects of buildings, future versions of the program will attempt to address other issues that are important to the built environment: water, electrical systems, etc.

Although it is important to note what EnergyPlus is, it is also important to remember what it is not.

- EnergyPlus is not a user interface. It is intended to be the simulation engine around which a third-party interface can be wrapped. Inputs and outputs are simple ASCII text that is decipherable but best left to a GUI (graphical user interface). This approach allows interface designers to do what they do best—produce quality tools specifically targeted toward individual markets and concerns. The availability of EnergyPlus frees up resources previously devoted to algorithm production and allows them to be redirected interface feature development in order to keep pace with the demands and expectations of building professionals.
- EnergyPlus is currently not a life cycle cost analysis tool. It produces results that can then be fed into an LCC program. In general, calculations of this nature are better left to smaller “utility” programs which can respond more quickly to changes in escalation rates and changes to methodologies as prescribed by state, federal, and defense agencies.
- EnergyPlus is not an architect or design engineer replacement. It does not check input, verify the acceptability or range of various parameters (expect for a limited number of very basic checks), or attempt to interpret the results. While many GUI programs assist the user in fine-tuning and correcting input mistakes, EnergyPlus still operates under the “garbage in, garbage out” standard. Engineers and architects will always be a vital part of the design and thermal engineering process.

Why does EnergyPlus exist and what were its original goals?

The existence of EnergyPlus is directly related to some of the increasingly obvious shortcomings of its predecessor programs—BLAST and DOE–2. Both programs, though still valid tools that will continue to have utility in various environments, have begun to show their age in a variety of ways. Both BLAST and DOE–2 were written in older version of FORTRAN and used features that will eventually be obsolete in new compilers. Both programs consisted of a significant amount of “spaghetti code” and outdated structures that made it

difficult to maintain, support, and enhance. Neither BLAST nor DOE–2 is able to correctly handle feedback from the HVAC system to the zone conditions.

Finally, the speed with which new technology in the HVAC field is developed has far outpaced the ability of the support and development groups of both programs to keep the programs current and viable. This is really the key issue in the existence of EnergyPlus: there simply are not enough researchers worldwide who have enough experience with the complex code of the programs to keep pace with new technology. In addition, due to the years of experience necessary to make modifications to either BLAST or DOE–2, it is extremely expensive and time consuming to produce models or train someone to become proficient in either program's code. The following paragraphs address how the goals of EnergyPlus have been carefully selected to solve the problems of the legacy programs and to make EnergyPlus an international repository for algorithms that can be made available to all interested parties.

Structure and Simulation Management: The "spaghetti code" nature of the parent programs resulted in great confusion as to how information flowed within the program, when data was modified or should be modified, and what was happening when. Thus, one of the goals of EnergyPlus development was to eliminate the interconnections between various program sections and the need to understand all parts of the code just to make an addition to a very limited part of the program. This goal was achieved through a program structure that implemented a "manager" philosophy. In effect, the program consists of many pieces (i.e., modules—see next paragraph) which when viewed graphically resemble an inverted tree. Instead of allowing lower level pieces of the program and calls to those sections of the code to infiltrate up to the highest level of the simulation, each branch exercises control over when its sub-branches are called. Thus, this is much like an organization chart where management over employees proceeds up a chain of command until it reaches the "CEO" of the simulation. In EnergyPlus, a subroutine called ManageSimulation serves the "CEO" role. This single subroutine controls four of the five main simulation loops (environment, day, hour, sub-hour time step; the system time step is controlled by the HVAC manager). In addition, this subroutine also sets global flags that are used by other subroutines and modules to determine the state of the simulation so that each part of the program knows whether to read input, initialize, simulate, report, etc. The manager philosophy makes the links between program modules simple and explicit. Typically, each module is accessible only through a call to its manager routine. Developers no longer have to worry about hidden and unexpected connections between and among program subroutines.

Modularity: One of the benefits of the structural improvements over the legacy programs is that the code can now be much more object-oriented and modular in nature. A result of the goal for improved structure in EnergyPlus was the definition of a well-organized, module concept that would facilitate adding features and links to other programs. It was decided that FORTRAN90 would be used for the initial release of EnergyPlus due to the fact that it:

- is a modern, modular language with good compilers on many platforms
- allows C-like data structures and mixed language modules
- provides structure that begins to be object-based
- allows long variable names (up to 32 characters)
- provides backward compatibility with legacy code during the development process

The key benefit of modularity is that researchers can develop modules concurrently without interfering with other modules under development and with only a limited knowledge of the entire program structure. This feature is critical to promoting both widespread use and broad development of program models. EnergyPlus can only be successful if the pool of potential model developers increases significantly so that interest and economics drives what models are added to the program rather than have its extensions limited by a lack of intellectual

resources. Thus, modularity and structure improvements in EnergyPlus solve the problems of not having enough experts to keep up with technology and the large start-up time and expense previously needed to bring someone up to speed on program additions.

Established Links: The modularity of EnergyPlus makes it easier for other developers to quickly add other component simulation modules. This means that it will be significantly easier to establish links to other programming elements. Since initially the EnergyPlus code will contain a significant number of existing modules, there will be many places within the HVAC code where natural links to new programming elements can be established. These are fully documented to assist other developers in a swift integration of their research into EnergyPlus. In addition to these more natural links in the HVAC section of the code, EnergyPlus will also have other more fluid links in areas such as the heat balance that will allow for interaction where the modules might be more complex or less component based. Again, the goal of this feature of EnergyPlus is maximize the number of developers who can quickly integrate their work into EnergyPlus for the minimum investment of resources. The following diagram depicts how other programs have already been linked to EnergyPlus and a big picture view of how future work can impact the program.

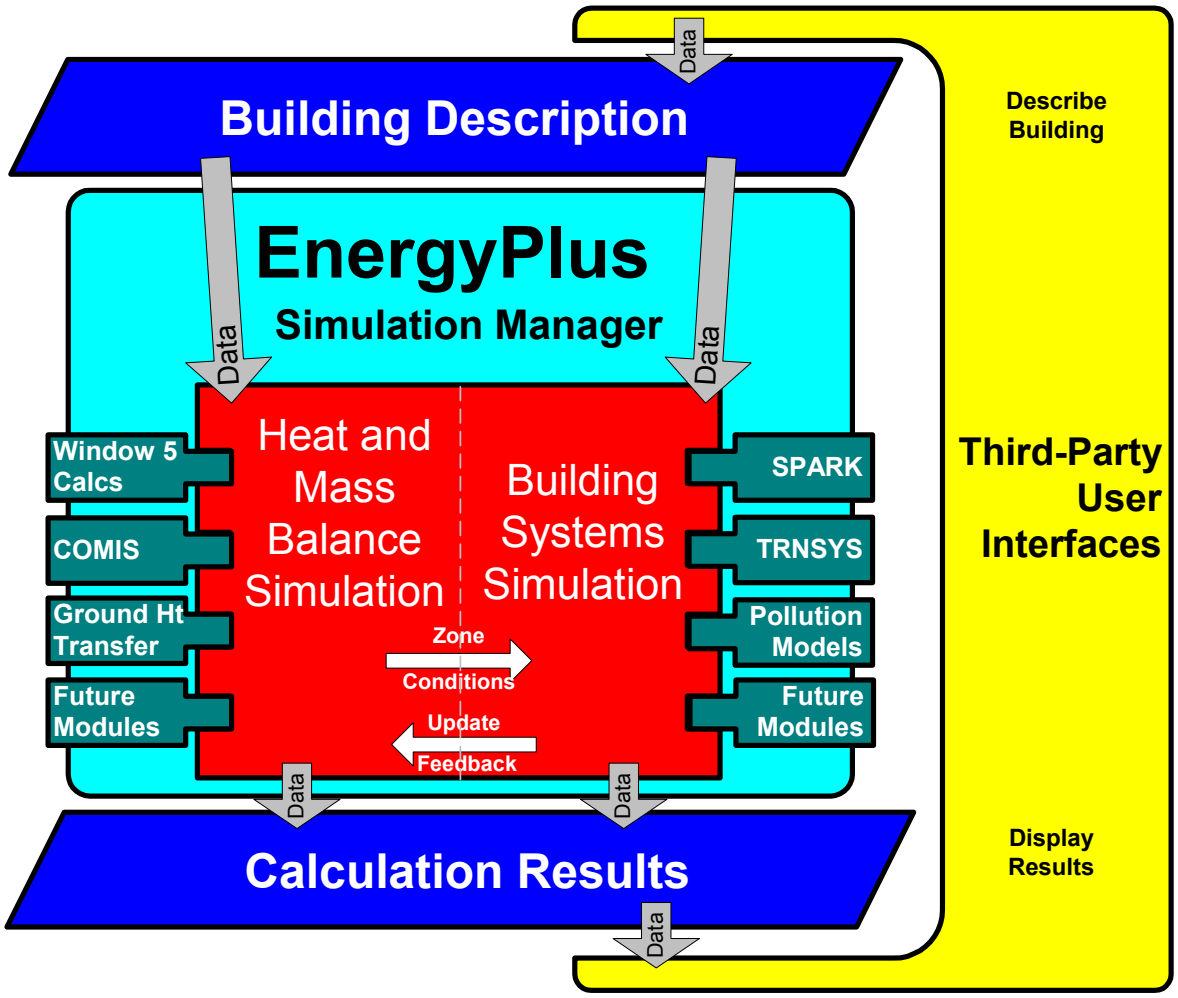

Figure 1. EnergyPlus -- the big picture

Integration of Loads, Systems, and Plants: One of the strong points of EnergyPlus is the integration of all aspects of the simulation—loads, systems, and plants. Based on a research version of the BLAST program called IBLAST, system and plant output is allowed to directly impact the building thermal response rather than calculating all loads

first, then simulating systems and plants. The simulation is coupled allowing the designer to more accurately investigate the effect of undersizing fans and equipment and what impact that might have on the thermal comfort of occupants within the building. The diagram below shows a basic overview of the integration of these important elements of a building energy simulation.

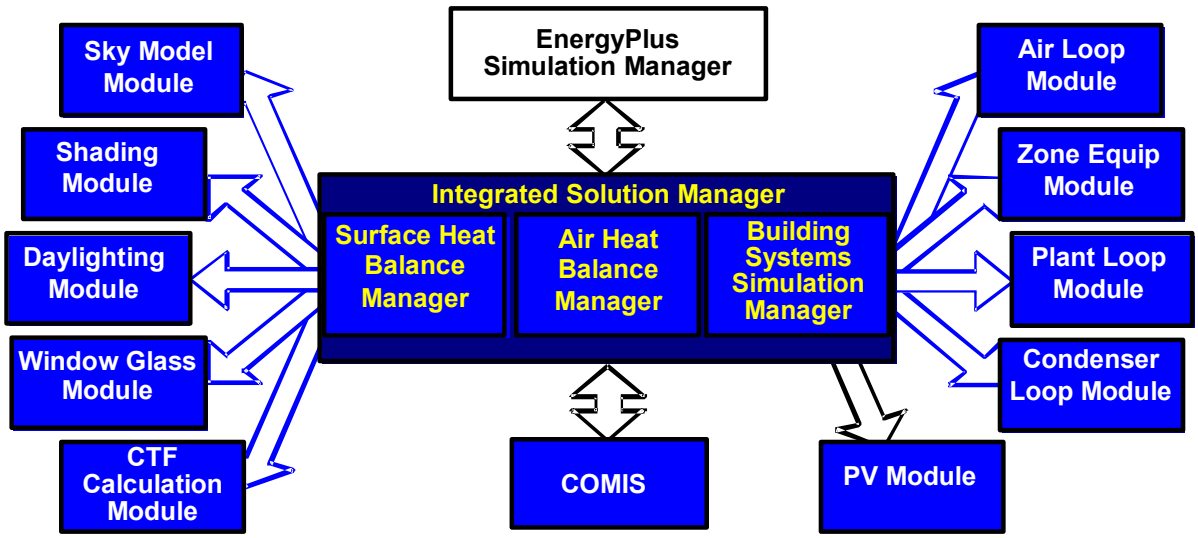

Figure 2. EnergyPlus -- Internal elements

"Open" Source Code: Another advantage of EnergyPlus that it shares with both BLAST and DOE–2 is that the source code of the program will be available and open for public inspection, revision, etc. The program is not intended to be a black box that is unintelligible to the users and developers around the world. While there are many conflicting ideas on what is meant by "open", EnergyPlus is certainly not closed since this would be counter to the goals that have already been stated. The hope is that this access to source code will improve the accuracy and usability of the program over the long term and allow many developers to work on the program simultaneously. This "developer friendly" stance is critical to keeping EnergyPlus current and in step with technological advances.

In summary, the goals of EnergyPlus are ambitious but achievable via the path described above. EnergyPlus aims to be a program that is relatively simple to work with from the perspective of both the users and the developer. The development team made tremendous efforts to keep simulation code and algorithms as separate as possible and as modular as possible to minimize the overall knowledge that someone would need to have to add models to the program. This will minimize the resource investment and maximize the impact of current research in the field of building energy analysis and thermal load calculations. Finally, the full coupling of building envelopes, systems, and plants will provide a better understanding of how a building responds not only to the environmental factors that impact the building but also the HVAC system as it attempts to meet the thermal loads on the building.

It is also important to note that testing and verification are key issues in the development of any new program such as EnergyPlus. While there are large sections of EnergyPlus that consist of brand new code, the majority of the heat balance code can be traced back to the original parent programs. It should be noted that while this code has been significantly reengineered the team used what has been termed an "evolutionary reengineering" (ER) process. In ER, program code is modified stepwise in an effort to restructure it, modularize it, replace old obsolete data structures, etc. The ultimate goal is to bring it up to current programming standards without starting over with new code. At each step along the way, the program is exercised over a variety of input files and parameters to insure that what were intended to be algorithm neutral changes in the program have not resulted in changes to the

output. This process was very successful and bolstered confidence in the program. In addition, comparisons could be made back to legacy programs to show that the new program is at a minimum as accurate as its predecessors. Beyond this, EnergyPlus has also been subjected to a lengthy and rigorous testing plan by an outside agency as well as numerous beta testers. This level of effort and collaboration is unprecedented in the history of energy analysis and thermal load calculation programs and has resulted in a much higher level of confidence in the results produced by EnergyPlus.

EnergyPlus Documentation Library

Below are brief descriptions of the various components of the EnergyPlus Documentation Library.

User Information Documents

The following documents relate to using EnergyPlus, the engine. These documents cover a full range of questions and should be the first place a beginning or even experienced user would go to find out how the program works, what it expects as input, what it produces as output, etc. In general, the information in these documents is not highly technical, but it is detailed enough to use the basic capabilities of the program.

Getting Started with EnergyPlus: You are currently reading the Overview section of this document. The overview contains a "big picture" description of the EnergyPlus program as well as background of its development and the goals to which it ascribes. The remainder of the Getting Started document provides beginning users with an introduction into how to run EnergyPlus, what files are needed for EnergyPlus to execute, and what files are produced when EnergyPlus runs successfully. It also provides some guidance as to how to determine what potential sources of errors are when EnergyPlus runs into problems and how serious those problems might be.

Input and Output Reference: This document is a thorough description of the various input and output files related to EnergyPlus, the format of these files, and how the files interact and interrelate. Included here is the description of weather data used in EnergyPlus.

Output Details, Examples and Data Sets: While the Input and Output Reference document touch on some of the possible outputs from EnergyPlus, this document has more details and specific examples. It also addresses the example files that are distributed with EnergyPlus as well as the reference data sets that are included.

***Auxiliary Programs*:** This document contains information for some of the auxiliary programs that are part of the EnergyPlus package. For example, this document contains the user manual for the Weather Converter program, descriptions on using Ground Heat Transfer auxiliary programs with EnergyPlus, HVAC Templates, and other assorted documents.

Engineering Reference Document

This document provides more in-depth knowledge into the theoretical basis behind the various calculations contained in the program. This reference includes more information on modeling equations, limitations, literature references, etc. The document contains the following information and is structured along the lines of the above illustration (Figure 2. EnergyPlus -- Internal elements).

Heat Balance Overview and Reference: This section describes the heat balance calculations that form the basis of the EnergyPlus building model. It includes descriptions of shadowing calculations and other pieces of the model.

HVAC Overview and Reference: This section contains a description of the loop-based approach used by EnergyPlus to model the HVAC systems: air loops, water loops, etc.

It includes a description of the higher-level managers that control the simulation flow as well as some information on the various components that can be linked together to comprise an HVAC system.

HVAC Branch Based Input Description: This section is a special extension of both the input document and the HVAC overview document. It contains more detail on the various HVAC input objects and how these different object link together to form an HVAC description. It contains vital information mainly for the interface developer but also provides users with an in-depth look at the inner workings of the loop approach adopted by EnergyPlus.

Engineering Reference Appendix: The appendix contains all other information that did not fit any of the previous categories listed above or was applicable across several major sections of code. Readers should search the three documents listed above first before consulting the reference appendix.

Developer Information Documents

The following documents will be most useful to potential developers of EnergyPlus, both Interface Developers and Module Developers. Interface Developers will be creating input and output wraps on EnergyPlus so that is it is usable to the architect, design engineers, and others. Module developers will be creating new modules within the EnergyPlus structure and framework.

Programming Standards: This document is the source of all major policies relating to the actual EnergyPlus source code. It describes rules to which all program code should adhere, notes how to structure and modularize code, and provides a listing of "approved" input and output units. As a result, it is an important document for both interface and module developers.

Interface Developer's Guide: This document is critically important to persons interested in developing an interface that provides input to and read output from EnergyPlus. It is a comprehensive guide to the input data dictionary and the input data files that contain a user's building data. Each piece of input syntax is described in detail. In addition, the mechanism for obtaining output and the format in which output will be produced are discussed. This document also contains sections on weather files and units. Numerous samples and examples are given throughout the document with a full file length example provided in the appendix.

Module Developer's Guide: This document contains a wealth of information that is intended to provide as much assistance as possible to persons interested in adding modules to the EnergyPlus program. It reviews the module concept as outlined in the programming standard and how they have been implemented in EnergyPlus. It provides a description of how the various modules work together and how the program is structured from a module tree (inverted tree) perspective. One of the most important features of this document is a list of standard EnergyPlus service subroutines and modules that greatly simplify the developers' task of integrating their work into the program. Input and output issues are also addressed from the perspective of how modules actually obtain data from the input file and how each section of the code sends data to the output files. The appendices of this document list code for two "global" data modules that are available to all parts of the code through the FORTRAN90 "USE" statement.

FAQ Document

The Frequently Asked Question / Commonly Asked Questions document contains questions and answers about EnergyPlus.

Getting Started with EnergyPlus

The standard install procedure has put the following information on your computer, in the directories shown:

(You were allowed to select components, so all of these may not be there).

\EnergyPlus

the main execution area, contains **Energy+.idd**, **EnergyPlus.exe**, **RunEPlus.bat**, shortcut to **IDFEditor**, **readme file(s)**, **EP-Macro.exe**, **bugreprt.txt** file. This will also contain the translation.exe for converting last "release" input files to the current release (see: Auxiliary Programs document for details) and the WinEPDraw program (again in Other Information document). If chosen, the EP-Launch program will be in this directory as well.

\Documentation

the PDF files of the documentation

\DataSets Reference Data Sets.

\MacroDataSets Data sets that are "macroized"

\PreProcess

\IDFEditor Program files for the IDFEditor

\BLASTTranslator

RunXLate batch file for executing the BLASTTranslator

Sample file described in this section (both BLAST versions and IDF versions)

\Libraries

the executables and libraries for the BLASTTranslator

\DOE2Translator

\IFCtoIDF Special program to convert IFC compliant files (CADD, etc) to EnergyPlus input format.

\GrndTempCalc Special program to calculate ground temperatures.

\WeatherConverter

EnergyPlus weather conversion/reporting program. Can transform files into EnergyPlus (EPW) format as well as Comma-delimited (CSV) and small statistical reports.

\PostProcess

ReadVarsEso The simple post processor exe.

\ExampleFiles Sample input, output, results files shipped with the program.

\WeatherFiles Sample weather files shipped with the program.

\Templates Input templates to assist in HVAC file creation.

Running EnergyPlus

Introduction

EnergyPlus is a simulation program designed for modeling buildings with all their associated heating, ventilating, and air conditioning equipment. EnergyPlus is a simulation *engine*: normally it would be used in conjunction with a graphical user interface. When graphical interfaces become available, input files will typically be created using these interfaces. For testing and development purposes, however, it can be run *stand alone* without such an interface. This document describes how to run EnergyPlus in such a stand alone fashion. If you are not familiar with building modeling and simulation programs, please review the section “Information necessary to Run EnergyPlus” first and then come back to this section.

Like all simulation programs, EnergyPlus consists of more than just an executable file. EnergyPlus needs various input files which describe the building to be modeled and the environment surrounding it. The program produces several output files, which need to be described or further processed in order to make sense of the results of the simulation. Finally, even in stand-alone mode, EnergyPlus is usually not executed “by hand”, but rather by running a procedure file which takes care of finding input files and storing or further processing the output files.

To assist those not familiar with the DOS Command Line or batch files, we have included the EP-Launch file. Review the next section for instructions. If you wish to learn about DOS/Command Line use for EnergyPlus, skip to section “RunEPlus batch file”.

EP-Launch Program

EP-Launch is an optional component of the EnergyPlus installation. For users that want a simple way of selecting files and running EnergyPlus, EP-Launch provides this and more. In addition, EP-Launch can help open a text editor for the input and output files, open a spreadsheet for the postprocessor results file, and start up a viewer for the selected drawing file.

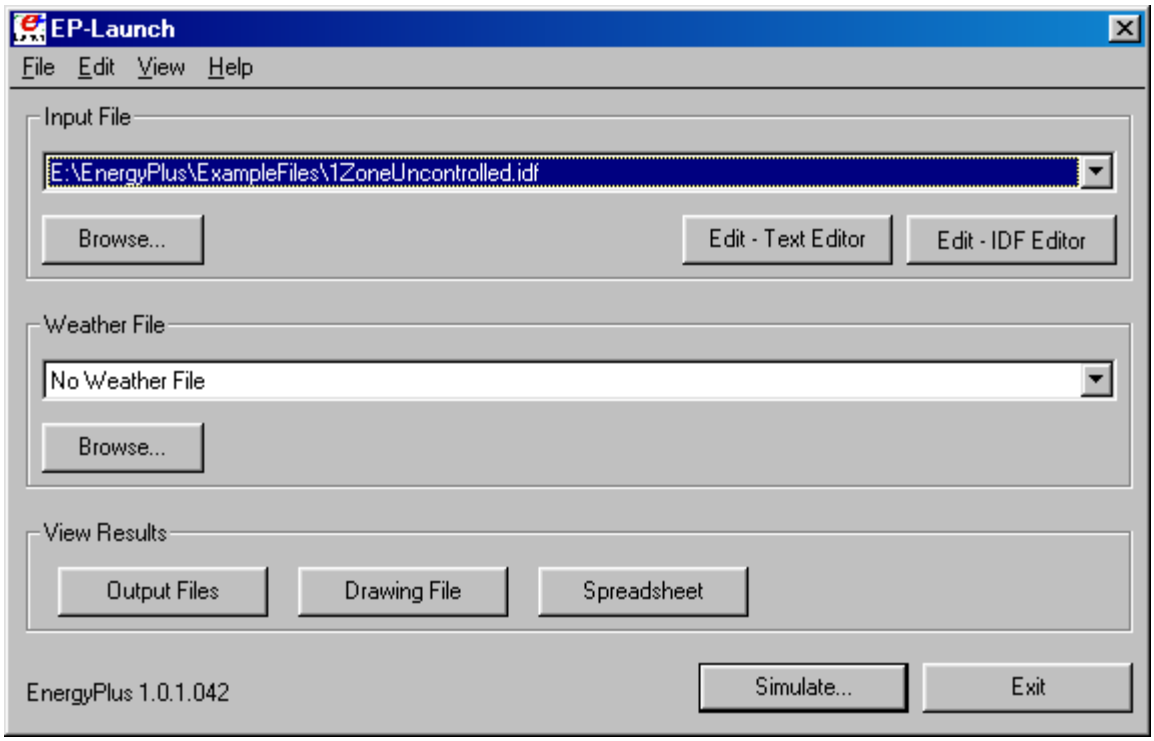

Figure 3. EP-Launch Screen

Start EP-Launch

EP-Launch should be located in the directory that you installed EnergyPlus. By double clicking on the EP-Launch icon you get the screen shown above. The EP-Launch program simply starts other programs and allows you to avoid having to use the DOS command line prompt to run EnergyPlus. More help is provided for the program under the “Help” menu.

Selecting Input and Weather Files

The input file and weather files can be selected from the two pull down lists which show recently used files or you can press the "Browse…" buttons to locate an input or weather file that you have created yourself.

Running a Simulation

After you select the weather and input files simply push the "Simulate…" button to start the EnergyPlus building energy simulation engine. At this point a black DOS window should pop up on your screen and show the progress of your simulation. The simulation is complete when the black DOS box closes. The EnergyPlus program black DOS window will show scrolling text as the simulation procedure progresses. If you would like to see these messages more slowly you have two options:

1) Press the "Control-S" key combination to try to stop the progress and any key to continue.

2) Under the "View" menu on the EP-Launch program, select "Pause During Simulation" and this will insert a pre-programmed pause immediately after EnergyPlus executes. To continue after the pause, press any key.

Looking at the Results

After you have run a simulation and the black DOS window closes, EnergyPlus has completed, and a status message is displayed:

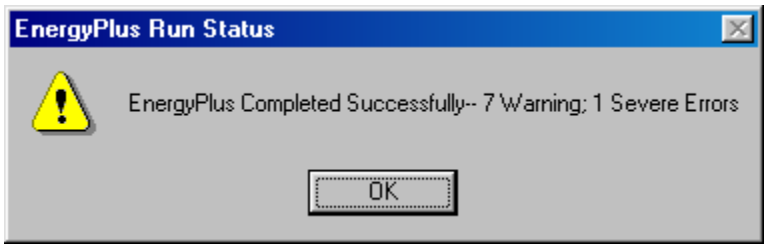

Figure 4. EP-Launch Finish Status.

After pressing OK on this box, you can press the F2 function key to display the .err, .eio, and .bnd files – useful when errors may have occurred.

You can view the results using one of the three buttons ("Output Files," "Drawing Files" and "Spreadsheet") in the "View Results" area of the screen.

By pressing the "Output Files…" button, a text editor will open each of the text output files. Up to 16 files will open, if they exist. Each file may also be opened individually from the View menu or with a function key. The output files and function key shortcuts are listed below:

1. ESO –raw report variable output (F5),
2. RDD –list of output variables available from the run (F6).
3. EIO –additional EnergyPlus results (F7),
4. ERR – list of errors and warnings (F8),
5. BND – HVAC system node and component connection details (F9),
6. MTR – raw report meter output (F11),
7. MTD – list of meter component variables (F12)
8. ZSZ – zone sizing details (Ctrl+F5)
9. SSZ – system sizing details (Ctrl+F6)
10. TRN – TRNSYS errors and warnings for photovoltaics (Ctrl+F7)
11. AUDIT – input file echo with input processors errors and warnings (Ctrl+F8)
12. SLN – output from "report, surfaces, lines" (Ctrl+F9)
13. DBG – output from the debug command (Ctrl+F11)
14. CIF – COMIS input file (Ctrl+F12)
15. EPMIDF – clean idf file after EP-Macro processing (Shift+F5)
16. EPMDET – EP-Macro detailed output with errors and warnings (Shift+F6)

The "Drawing File" and "Spreadsheet" buttons will open the generated DXF and CSV files respectively, only if appropriate viewers have been configured, see *Selecting Viewers and Editors* below. The CSV file contains the same results as the ESO file described above but in a tabular fashion appropriate for a spreadsheet. The <filename>Meter.CSV file contains tabulr results from the MTR file. The DXF file is a CAD format that displays the physical shape of the building being modeled in three dimensions.

Viewing the Drawing File without Running a Simulation

The "Drawing" button (or the View menu Drawing File option) will automatically run WinEPDraw if the DXF file does not exist or it is older than the input file. This allows the building geometry to be viewed without running a full simulation. For more information about WinEPDraw, see the *Auxiliary EnergyPlus Programs* document.

Editing the Input Files

The input file, called IDF file that is selected from the top pull-down list, can be edited by pressing one of two buttons in the "Input File" area. The "Edit - Text Editor" button will start a text editor and the "Edit - IDF Editor" will start the separate program called the IDF Editor.

Remember to save any changes you make in either editor before returning to EP-Launch to run the simulations again.

Selecting Viewers and Editors

The default text editor that the EP-Launch program will start is the application associated with .txt files, usually NOTEPAD.EXE. This can be changed to your favorite text editor by going to the "File" menu in EP-Launch and clicking on "Select Text Editor…" then locating the executable file for your favorite editor.

The default spreadsheet program is the application associated with .csv files, such as Microsoft Excel. This can be changed to your favorite spreadsheet program by going to the "File" menu in EP-Launch and clicking on "Select Spreadsheet Program…" then locating the executable file for your favorite spreadsheet software.

The default drawing viewer is the application associated with .dxf files. This can be changed to your favorite spreadsheet program by going to the "File" menu in EP-Launch and clicking on "Select Drawing Viewer …" then locating the executable file for your favorite drawing software capable of reading a dxf file. AutoDesk's VoloView Express may be used as the DXF file viewer. It is available for free from AutoDesk's web site.

The selection of viewers is saved between times that you use the EP-Launch application. If one of the file types (txt, csv, or dxf) does not have an associated application, you will be prompted to select a program exe when you try to view or edit that file type. To reset the default applications, select "Autofind Programs" from the File menu.

File Menu

The File menu can be used for selecting input and weather files just like the "Browse…" buttons (see the *Selecting Input and Weather Files* section above). In addition, the File menu is used to select the appropriate file viewers/editors (see the *Selecting Viewers and Editor* section above) and can be used to run a simulation.

Edit Menu

No cutting or pasting is used in this program so the edit menu shows options that duplicate the functions of the "Edit – Text Editor" and "Edit – IDF Editor" buttons. In addition, the weather file and the postprocessor command file (rvi) may be opened in the text editor.

View Menu

The View menu duplicates the options in the "View Results" area of the main screen (see the *Looking at the Results* section above) and allows opening of selected output files. In addition, you can turn on and off the "Pause During Simulation" feature described in the *Running a Simulation* section above, select a wide format (useful for long path names), or select an alternative interface with a different arrangement of buttons.

Help Menu

The Help menu can be used to open the EnergyPlus documentation files and the EP-Launch help file.

Recently Used Files

The recently used input and weather file pull down lists can hold a maximum of twenty items. These lists, like the viewers selected, are saved between times you use the EP-Launch program.

Caveats

Remember to save changes made in the editor before you run another simulation.

You will need to close the spreadsheet program that views the resulting CSV file prior to another simulation and you may need to close the text editor windows also (depending on your editor).

The EPL-RUN.BAT batch file is used to run EnergyPlus from the EP-Launch program. It can be edited with care if other postprocessors or preprocessors are to be used.

When things go wrong

Though EnergyPlus has had several releases (including beta releases prior to initial release), there still may be problems when input files meet with EnergyPlus. If you are using EP-Launch when this happens, you will see a window appear as in the figure below. Follow the instructions listed on the screen.

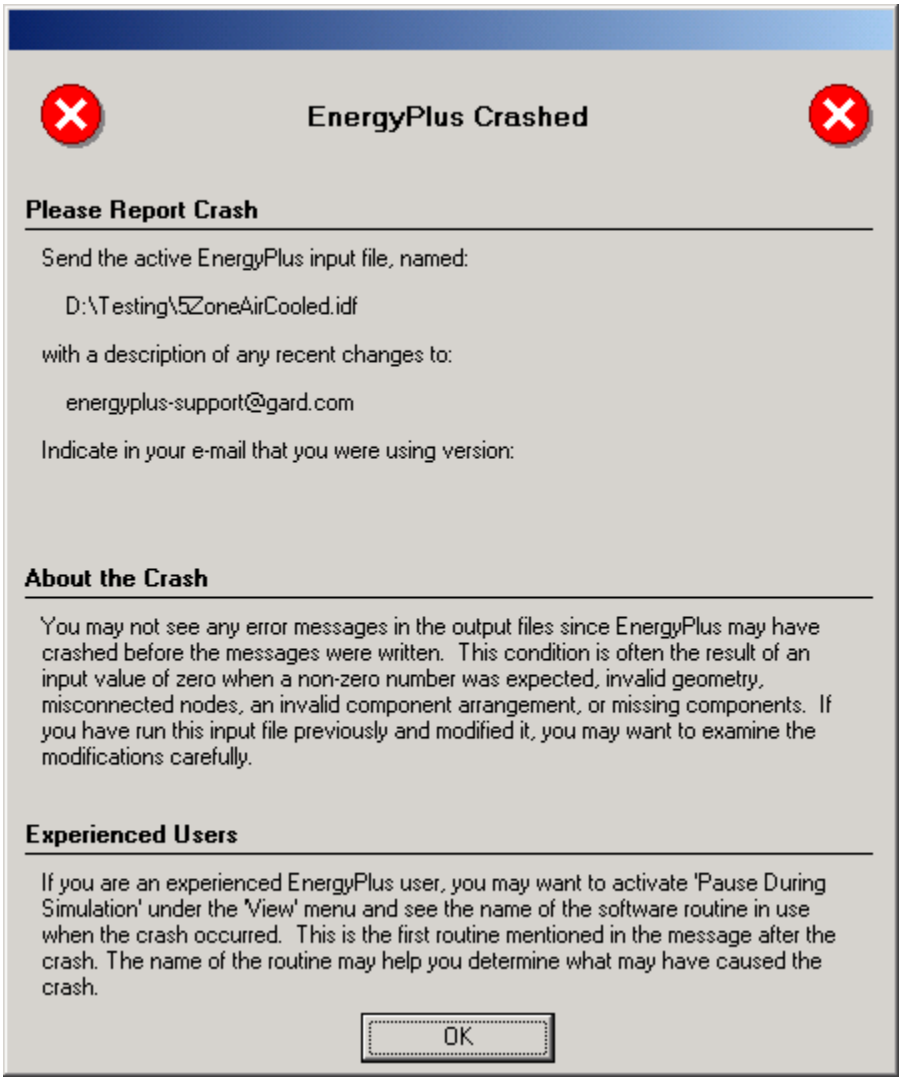

Figure 5. EnergyPlus crash within EP-Launch

Bugs

The EP-Launch program has been through several “releases” though there is a good chance you will find bugs. Please report them to the energyplus-support@gard.com address so that we can fix them prior to the beta release.

IIf the pull down lists ever are shown as blank the "reset" button may be used. This unlabeled button is very small in the lower left-hand corner of the main screen. It removes the items

shown in the recently used file list and causes the program to forget the viewers and text editors selected previously and exits the program. When you start EP-Launch again, you will need to make these selections again.

RunEPlus batch file

It is simple to run EnergyPlus: open a DOS or COMMAND box in the EnergyPlus directory and use the RunEPlus.bat procedure file. The procedure file takes two arguments – the input file name (containing the building description) and the weather file name:

RunEPlus <input_file_name> <weather_file_name>

Though it's possible to supply arguments to the batch file with embedded blanks, it's better practice not to. More extensive information about the intricacies of EnergyPlus execution is given in the next two sections of this document.

As installed, the RunEPlus is ready to run the sample files that are included. If you wish to create and run other files, you may wish to modify the batch file to your own preferences. See "Running EnergyPlus Using a procedure file" later in this document for parts of the batch file that you will need to modify.

Running EnergyPlus by Hand

EnergyPlus is compiled as a 32 bit console application on Windows™ (Windows 95, Windows 98, Windows NT, Windows 2000, Windows ME) operating systems, commonly run on the Intel™ or compatible processing chips (aka WinTel machines). To run the program bring up the command prompt and "cd" to the directory containing the executable. Assume that the executable is called *EnergyPlus.exe*. In the same directory EnergyPlus expects *in.idf*, the input data file; *Energy+.idd*, the data dictionary file; *in.epw*, the weather file (needed only if there is a RunPeriod in the input); and optionally *Energy+.ini*, the initialization file. Typing "EnergyPlus" (and hitting the *Enter* key) will execute the program. EnergyPlus will write messages to the command window as it runs. A simulation with two design days and one run period looks like:

```
EnergyPlus Starting
Warming up
Initializing Response Factors
Matrix exponential for Construction          1
Invert Matrix for Construction         1
Gamma calculation for Construction          1
Compute CTFs for Construction         1
Matrix exponential for Construction          2
Invert Matrix for Construction         2
Gamma calculation for Construction          2
Compute CTFs for Construction         2
Matrix exponential for Construction          3
Invert Matrix for Construction         3
Gamma calculation for Construction          3
Compute CTFs for Construction         3
Matrix exponential for Construction          3
Invert Matrix for Construction         3
Gamma calculation for Construction          3
Compute CTFs for Construction         3
Matrix exponential for Construction          4
Invert Matrix for Construction         4
Gamma calculation for Construction          4
Compute CTFs for Construction         4
Initializing Window Optical Properties
Initializing Solar Calculations
Warming up
Warming up
Warming up
Performing Simulation
Warming up
Initializing Solar Calculations
Warming up
```

```
Warming up
Performing Simulation
Warming up
Initializing Solar Calculations
Warming up
Warming up
Warming up
Performing Simulation
EnergyPlus Completed Successfully.
```

When execution is finished, *eplusout.err* and *audit.out* will always appear. If the program terminated with an input error, these will be the only output files. If the program runs normally, *eplusout.eio* will appear. Depending on what was requested in the input, the other output files described above will also be written.

Running EnergyPlus Using a procedure file

A procedure (batch) file is the normal way to run a console application. The *installed* procedure file **RunEPlus.bat** can be used to execute EnergyPlus and deal with all the file handling and postprocessing. It can also accommodate running the EPMacro program if you name your files appropriately.

The "set" statements near the beginning of the procedure file should be customized for each local system. Thus "program_path" should be set to the directory path where the program executable resides on your local computer, "program_name" should be set to the name of the EnergyPlus executable file, "input_path" should be set to the directory path containing the input (IDF) file, and so forth. The batch file is executed by typing:

```
RunEPlus <input_filename> <weather_filename>
```

where <input_filename> is the name of the IDF file, without the file extension, and <weather_filename> is the name of the weather file, without the file extension.

EnergyPlus Files

Input Files

The following files are input to the EnergyPlus program.

Energy+.idd

The *input data dictionary* (IDD) is an ascii (text) file containing a list of all possible EnergyPlus objects and a specification of the data each object requires. This file is analogous to the DOE-2 keyword file. The *Guide for Interface Developers* contains a full description of the input data dictionary.

in.idf

The *input data file* (IDF) is an ascii file containing the data describing the building and HVAC system to be simulated. The *Guide for Interface Developers* shows examples of IDF input. Many example files are installed as part of the EnergyPlus installation.

Energy+.ini

This is the EnergyPlus initialization file. It is an optional ascii input file that allows the user to specify the path for the directory containing Energy+.idd. This file, using the actual directories of the install, will be created during the install. An example is:

```
[program]
dir=C:\EnergyPlus
```

```
[weather]
dir=

[trnsys]
dir=C:\EnergyPlus\TRNSYSPV\
```

The "weather" portion of the initialization file is unused. The [trnsys] item points to the folder where the TRNSYS libraries for simulations using the TRNSYS features are stored.

in.epw

The *EnergyPlus weather* file is an ascii file containing the hourly or sub-hourly weather data needed by the simulation program. The data format is described in the *Guide for Interface Developers* and the *Other Information* document – both of these are under the *Enhancing EnergyPlus* Tab of the EnergyPlus Documentation Menu. It is also described in the Encyclopedic Input Output Reference document.

Output Files

More information (and more up-to-date) about output files is shown in the Output Details and Examples Document.

eplusout.err

A text file containing the error messages issued by EnergyPlus. This is the first output that should be examined after a simulation. Error messages may be issued by EnergyPlus during its input phase or during the simulation. There are three levels of error severity: *fatal*, *severe*, and *warning* as well as simple *"message"* lines. A fatal error causes the program to terminate immediately. The following table illustrates the necessary actions.

Table 1. Error Message Levels – Required Actions

Error Level	Action
"Message"	Informative, usually a follow-on to one of the others. No action required.
Warning	Take note. Fix as applicable.
Severe	Should Fix
Fatal	Program will abort

An example of an error message due to an input syntax error is:

```
** Severe  ** Did not find " DessignDay" in list of Objects
   **  Fatal  ** Errors occurred on processing IDF file -
       probable incorrect IDD file. View "audit.out" for details.
   ************* EnergyPlus Terminated--Error(s) Detected.
```

audit.out

This is an text file which echoes the IDD and IDF files, flagging syntax errors in either file. Note that both *eplusout.err* and *audit.out* will show the error messages caused by input syntax errors; however only *eplusout.err* will show errors issued during the actual simulation.

Audit.out can be used when you need to see the context of the error message to fully ascertain the cause.

eplusout.eso

The *EnergyPlus Standard Output* (ESO) is a text file containing the time varying simulation output. The format of the file is discussed in the *Guide for Interface Developers* and the *InputOutputReference*. The contents of the file are controlled by *Report Variable* commands in the IDF file. Although the ESO is a text file, it is not easily interpretable by a human. Usually postprocessing will be done on this file in order to put it in a format that can be read by a spreadsheet; however a quick visual inspection of the file does show whether the expected variables are output at the desired time step.

eplusout.mtr

The *EnergyPlus Meter Output* (MTR) is a text file containing the time varying simulation output. The format of the file is similar to the ESO file. As described in a later section of this document (see Energy Meters), meters are a powerful reporting tool in EnergyPlus. Values are grouped onto logical meters and can be viewed the same way that the ESO variables are used. The contents of the file are controlled by *Report Meter* commands in the IDF file. Although the MTR is a text file, it is not easily interpretable by a human. Usually postprocessing will be done on this file in order to put it in a format that can be read by a spreadsheet; however a quick visual inspection of the file does show whether the expected variables are output at the desired time step.

eplusout.eio

The *EnergyPlus Invariant Output* (EIO) is a text file containing output that does not vary with time. For instance, location information (latitude, longitude, time zone, altitude) appears on this file.

eplusout.rdd

The *Report (variable) Data Dictionary* (RDD) is a text file listing those variables available for reporting (on the ESO or MTR) for this particular simulation. Which variables are available for output on the ESO or MTR depends on the actual simulation problem described in the IDF. A simulation with no chiller would not permit the output of any chiller report variables. The user may need to examine the RDD to find out which report variables are available in a particular simulation. The RDD is written only if

```
Report, Variable Dictionary;
```

appears in the input (IDF) file.

eplusout.dbg

This is a text file containing *debug* output for use by EnergyPlus developers. Generally developers will add debug print statements wherever in the code that that they wish. There is a “standard” debug output that prints out conditions at all the HVAC nodes. This output is triggered by placing

```
DEBUG OUTPUT,1;
```

in the IDF file. If DEBUG OUTPUT, 0 is entered, you will get an empty eplusout.dbg file.

eplusout.dxf

This is a file in AutoCad DXF format showing all the surfaces defined in the IDF file. It provides a means of viewing the building geometry. The DXF file from EnergyPlus highlights different building elements (shading, walls, subsurfaces) in differing colors. A number of programs can read and display DXF files. One that works well is Volo View Express, available free from the Autodesk web site. Output of this file is triggered by

```
Report, Surfaces, DXF;
```

in the IDF.

eplusout.sln

A text file containing the coordinates of the vertices of the surfaces in the IDF.

Output of this file is triggered by

```
Report, Surfaces, Lines;
```

in the IDF.

Postprocessing Program/Files

A postprocessing program *ReadVarsESO.exe* is available that will read an ESO or MTR file and produce a file that can be read by Excel™. It can use an input file or not. In batch mode it is run by the little batch file *RunReadESO.bat*: Further information on this program is provided later in this document (see PostProcessing) and in the Input Output Reference as well as the Output Details and Examples documents.

Information necessary to Run EnergyPlus

Building Simulation

If you are already familiar with modeling buildings, particularly modeling buildings for energy consumption, you may wish to skip to “Creating Input Files”. The following steps have been liberally lifted and changed from the BLAST (Building Loads Analysis and System Thermodynamics) User’s Manual. Please excuse any inaccuracies in the translation from BLAST; these are meant as general guidelines.

A Methodology for Using Energyplus

This section provides a step by step outline that will help you streamline creating your building models for using EnergyPlus.

Step 1: Plan Ahead

Some preliminary steps will facilitate the construction of your input file. EnergyPlus requires some information in specified, externally available formats; other information may require some lead time to obtain. The following checklist should be completed before you start to construct your input file.

√ Obtain location and design climate information for the city in which your building is located. If possible, use one of the weather files available for your weather period run.

√ Obtain sufficient *building construction* information to allow specification of overall building geometry and surface constructions (including exterior walls, interior walls, partitions, floors, ceilings, roofs, windows and doors).

√ Obtain sufficient *building use* information to allow specification of the lighting and other equipment (e.g. electric, gas, etc.) and the number of people in each area of the building.

√ Obtain sufficient *building thermostatic control* information to allow specification of the temperature control strategy for each area of the building.

√ Obtain sufficient *HVAC operation* information to allow specification and scheduling of the fan systems.

√ Obtain sufficient *central plant* information to allow specification and scheduling of the boilers, chillers and other plant equipment.

Step 2: "Zone" the Building

A building "surface" is the fundamental element in the building model. In the general sense, there are two types of "surfaces" in EnergyPlus. These are:

1. heat transfer surfaces and
2. heat storage surfaces

The first rule of building modeling is, "*Always define a surface as a heat storage surface unless it must be defined as a heat transfer surface*". Any surface, which is expected to separate spaces of significantly different temperatures, must be defined as a *heat transfer surface.* Thus, exterior surfaces, such as outside walls, roofs and floors, are *heat transfer surfaces*. Interior surfaces (partitions) are *heat storage surfaces* if they separate spaces maintained at the same temperature and *heat transfer surfaces* if they separate spaces maintained at different temperatures. A discussion of how to define heat transfer and heat storage surfaces will occur in later steps. In order to correctly "zone" the building it is necessary only to distinguish between the two.

A "zone" is a *thermal,* not a *geometric,* concept. A “zone” is an air volume at a uniform temperature plus all the heat transfer and heat storage surfaces bounding or inside of that air volume. EnergyPlus calculates the energy required to maintain each zone at a specified

temperature for each hour of the day. Since EnergyPlus performs a zone heat balance, the first step in preparing a building description is to break the building into zones. The objective of this exercise is to define as *few* zones as possible without significantly compromising the integrity of the simulation.

Although defining building zones is somewhat of an art, a few general rules will keep the new simulation user out of trouble. Consider the following figure, which shows the floor plan of the Ft. Monmouth Education Center.

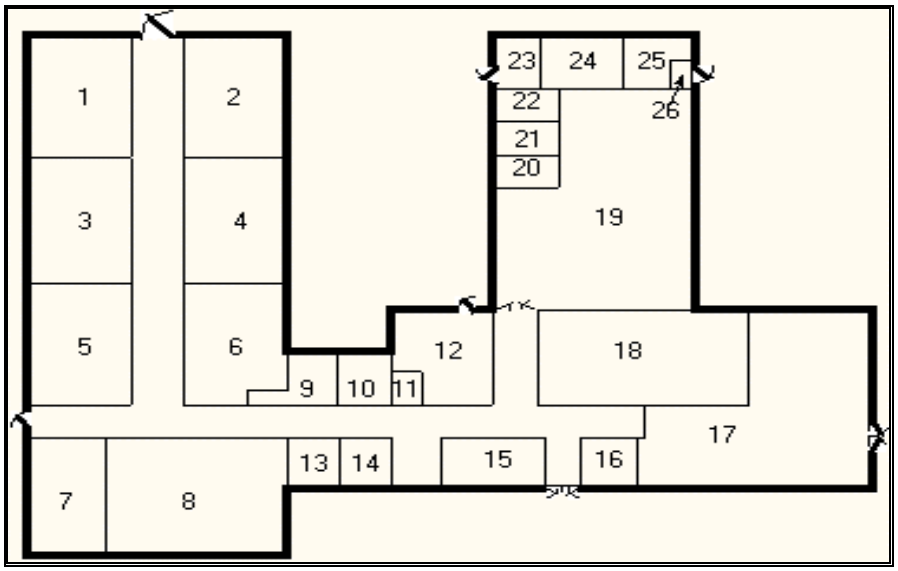

Figure 6. Fort Monmouth Education Center

The question is, "How many *thermal* zones should be used to model this building?" The inexperienced building modeler may be tempted to define each room in the building as a zone, but the thermal zone is defined as a volume of air at a uniform temperature. The general rule then is to *use the number of fan systems (and radiant systems) not the number of rooms to determine the number of zones in the building.* The minimum number of zones in a general simulation model will usually be equal to the number of systems serving the building. The collection of heat transfer and heat storage surfaces defined within each zone will include all surfaces bounding or inside of the space conditioned by the system.

Five systems were designed to serve the Ft. Monmouth Education Center. These systems with the thermal zones they serve are shown in the table below. The location of each zone is shown in accompanying figure.

System Number	System Name	CFM	Zone Served
1	Four Pipe Fan Coil	3900	Zone 1
1	Four Pipe Fan Coil	2500	Zone 2
2	Single Zone Draw Through	1400	Zone 3
3	Single Zone Draw Through	2250	Zone 5
4	Single Zone Draw Through	2450	Zone 6
5	Unit Heater	185	Zone 4
5	Unit Heater	41	Zone 7

Table 2. Zoning the Building by System Type

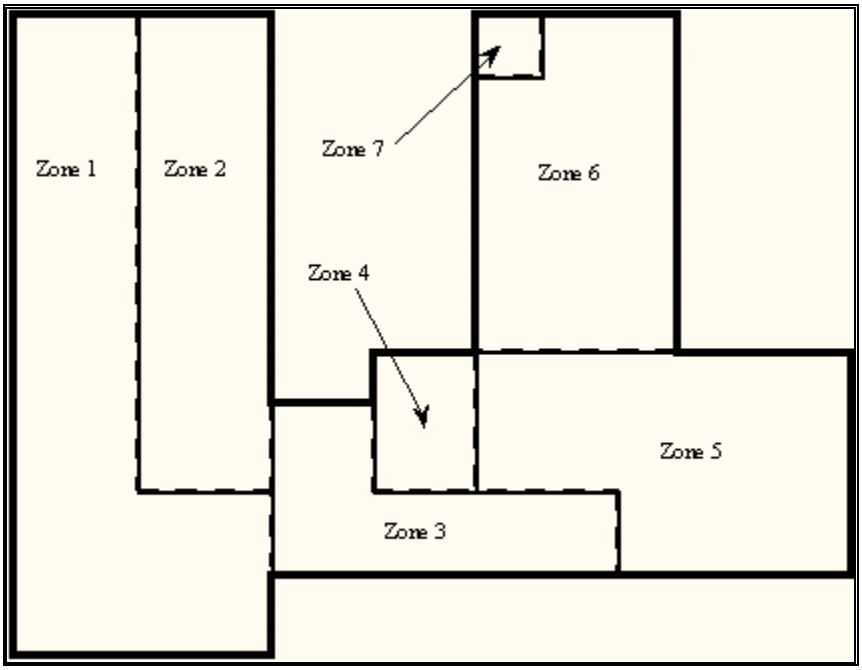

Figure 7. Thermal Zones in the Education Center

Notice Zone 1, Zone 2, Zone 4, and Zone 7. The two important zoning concepts can be demonstrated with the zoning of the Ft. Monmouth Education Center to reinforce the idea of a thermal zone and encourage the use of simplified models.

1. Notice that Zones 4 and 7 include two rooms that are not adjacent to one another but are served by the same system. Because the air temperature in the two spaces is maintained at the same uniform temperature, the two spaces, though separated spatially, may be defined as a single zone. For our purposes, we will define them as separate zones.

2. Notice that Zone 1 and Zone 2 are served by the same fan system and could be defined as a single zone with 7650 cfm of conditioned air supplied to the space. The space was split into two zones because the designer expected higher solar loads on the South and West sides of the wing and wanted to examine the *distribution* as well as the *magnitude* of the load in the space.

Zoning concept number 2 (above) leads to an important final point concerning the zoning of a building for a simulation. Complete estimates of the total building load (magnitude only) may be obtained with very simple models. For example the total building load calculated using a one-zone model of the Ft. Monmouth Education Center (Figure below) will **NOT** be significantly different from the total building load calculated using a more detailed model. The *distribution* of the load within the building cannot be estimated with the simplified building model, but its *magnitude* (such as would be used in sizing the central plant equipment) can be quickly estimated using a very simple model.

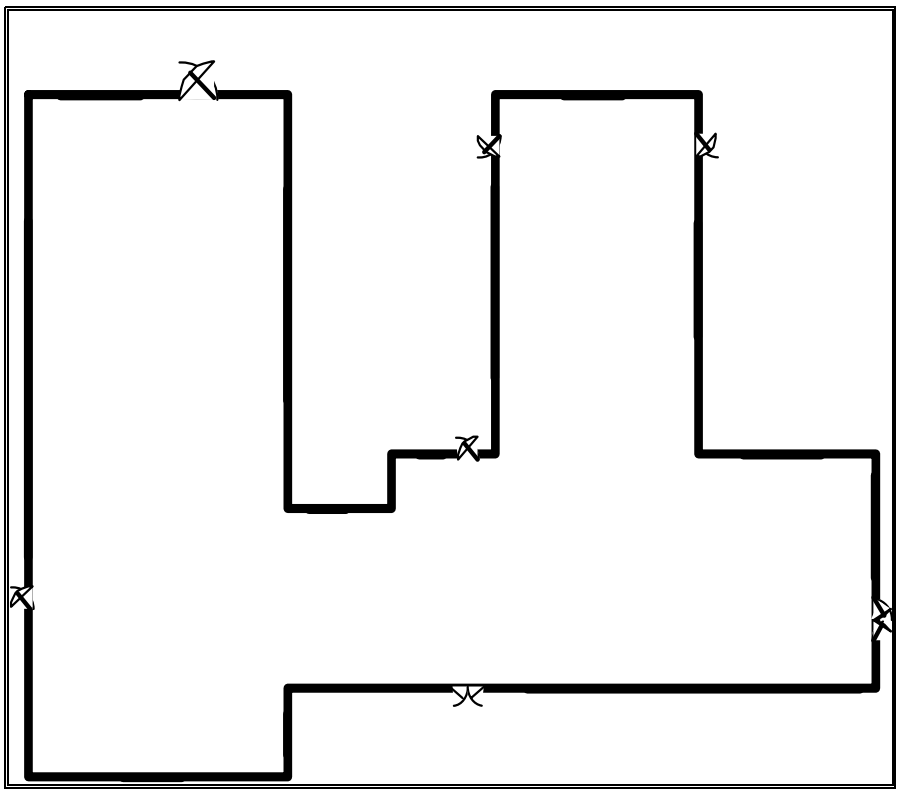

Figure 8. Single Zone Model of the Ft. Monmouth Education Center.

Step 3: Prepare to Construct the Building Model

Working from blueprints or sketches and following the guidelines in Step 2, the building zones were determined. It is recommended that the engineer sketch the building with its zones. Surface dimensions should be included in the sketch. Additional geometric and surface information is required before an input file describing the building can be constructed. Specifically the building model must:

1. Determine *heat transfer* and *heat storage* surfaces.
2. Define equivalent surfaces.
3. Specify surfaces and subsurfaces (windows, doors, etc.) construction and materials.
4. Compile surface and subsurface information.

At this point it is appropriate to begin to introduce the structure of the input file. Using the existing BLAST model, we have constructed the EnergyPlus input file by using the BLASTTranslator program. The single zone model, as described later in this section, will be included in the ExampleFiles when you install EnergyPlus. This file can be used as described in other parts of this document to perform the building analysis.

Step3.1. Determine heat transfer and heat storage surfaces.

The surfaces of the building can be described in any order; grouping surfaces by zone may help you read the input file. Specifics of the describing surfaces help categorize the surface's heat transfer/storage as well as identify the surface construction information.

The details of the inputting surfaces are described in the Input/Output Reference document. The allowable surface types are shown in the following table:

Table 3. Surface types and categorization

Surface Type	Applicability
Surface:HeatTransfer	Wall, Roof, Ceiling, Floor
Surface:HeatTransfer:Sub	Window, Door, Glassdoor
Surface:HeatTransfer:InternalMass	Areas internal to a zone
Surface:Shading:Detached	Shading devices external to the building face (other buildings, trees, etc.)
Surface:Shading:Attached	Shading devices attached to the building (overhang, fin)

The pieces of the definition that designate Surface:HeatTransfer surfaces as either *heat transfer* or *heat storage* surfaces are:

```
  A5 , \field OutsideFaceEnvironment
       \type choice
       \key OtherZone
       \key ExteriorEnvironment
       \key Ground
       \key OtherSideCoeff
  A6,  \field OutsideFaceEnvironment Object
       \type object-list
       \object-list ZoneNames
       \object-list OSCNames
  \note Used only if OutsideFaceEnvironment is OtherZone or OtherSideCoeff
  \note If OtherZone, specify name of corresponding surface in adjacent zone
       \note If OtherSideCoeff, specify name of OtherSideCoefficients
  A7 , \field Sun Exposure
       \type choice
       \key SunExposed
       \key NoSun
  A8, \field Wind Exposure
       \type choice
       \key WindExposed
       \key NoWind
```

Note that subsurfaces (windows, doors) on these base surfaces will inherit the base surface properties listed above. The following examples will use a bit more of the Surface definition to give context.

Surfaces that specify “themselves” as the exterior environment are ceilings, floors and partitions that divide temperature-controlled spaces. The program assumes that the surface temperatures on both sides of the surface are the same. This means that even though heat may be stored in a partition, ceiling, or floor, no heat flows *through* it.

Heat Storage Surfaces (Use current Surface name for ExteriorEnvironment), e.g.:

```
Surface:HeatTransfer,Zn005:Wall006,  !- Base Surface Name
  Wall,INTERIOR,  !- Class and Construction Name
  MAINE WING,  !- Zone
OtherZone, Zn005:Wall006,  !- Exterior Conditions and Target (if applicable)
   NoSun,  !- Solar Exposure
   NoWind,  !- Wind Exposure
  0.5000000    ,  !- VF to Ground
           4, !-Rectangle
   57.90000    ,   57.79000    ,   10.00000    ,
   57.90000    ,   57.79000    ,  0.0000000E+00,
   57.90000    ,   47.79000    ,  0.0000000E+00,
   57.90000    ,   47.79000    ,   10.00000    ;
```

Some surfaces divide the temperature controlled space from the outside environment. Surfaces that are both sun and wind exposed (e.g. exterior walls, exposed floors, roofs) feel the full effect of both solar radiation and outside temperature, and the outside air film resistance for these surfaces changes with wind speed and wind direction. Surfaces that are not sun or wind exposed (a wall to an "uncontrolled" space) are not affected by solar radiation, wind speed or direction and have a constant outside convective air film resistance.

Heat Transfer Surfaces Exposed to the Outside Environment, such as Exterior Walls, Roofs, Exposed Floors:

```
Surface:HeatTransfer,Zn005:Wall002,  !- Base Surface Name
  Wall,EXTERIOR,  !- Class and Construction Name
  MAINE WING,  !- Zone
  ExteriorEnvironment,,  !- Exterior Conditions and Target (if applicable)
   SunExposed,  !- Solar Exposure
   WindExposed,  !- Wind Exposure
  0.5000000    ,  !- VF to Ground
           4, !-Rectangle
   77.90000    ,   47.79000    ,   10.00000    ,
   77.90000    ,   47.79000    ,  0.0000000E+00,
   77.90000    ,   67.79000    ,  0.0000000E+00,
   77.90000    ,   67.79000    ,   10.00000    ;
```

Surfaces such as basement walls and slab floors separate the space from the earth surrounding the surfaces. Therefore, the outside surface temperatures become the ground temperatures.

Heat Transfer Surfaces in Contact with the Ground, such as Basement Walls or Slab Floors:

```
Surface:HeatTransfer,Zn004:Flr001,  !- Base Surface Name
  Floor,SLAB FLOOR,  !- Class and Construction Name
  ARIZONA WING,  !- Zone
  Ground,,  !- Exterior Conditions and Target (if applicable)
   NoSun,  !- Solar Exposure
   NoWind,  !- Wind Exposure
   1.000000    ,  !- VF to Ground
           4, !-Rectangle
   38.01000    ,   8.510000    ,  0.0000000E+00,
   18.01000    ,   8.510000    ,  0.0000000E+00,
   18.01000    ,   28.51000    ,  0.0000000E+00,
   38.01000    ,   28.51000    ,  0.0000000E+00;
```

Other surfaces separate zones that may be at different temperatures. These surface types allow heat transfer (by conduction through the walls) from a zone at a higher temperature to a zone at a lower temperature. The location of the heat storage surface in the zone is not important except in specialized solar studies. The surface above (wall to uncontrolled space) would be more correctly modeled as an interzone surface.

Heat Transfer Surfaces Exposed to Another Zone, such as Interzone walls, ceilings or floors:

```
Surface:HeatTransfer,Zn005:Wall005,  !- Base Surface Name
  Wall,INTERIOR,  !- Class and Construction Name
  MAINE WING,  !- Zone
  OtherZone,Zn001:Wall009,  !- Exterior Conditions and Target (if applicable)
   NoSun,  !- Solar Exposure
   NoWind,  !- Wind Exposure
  0.5000000    ,  !- VF to Ground
           4, !-Rectangle
   57.90000    ,   47.79000    ,   10.00000    ,
   57.90000    ,   47.79000    ,  0.0000000E+00,
   67.90000    ,   47.79000    ,  0.0000000E+00,
   67.90000    ,   47.79000    ,   10.00000    ;
```

Step 3.2. Define equivalent surfaces.

When the building was zoned our objective was to define as *few* zones as possible. Now we would like to extend this objective to include defining as *few* surfaces as possible without significantly compromising the integrity of the simulation. We reduce the number and complexity of surfaces in our input file by defining *equivalent* surfaces.

Before dealing with equivalent surfaces, it is appropriate to take the concept of a thermal zone one step further. EnergyPlus performs heat balances on individual zone surfaces and on the zone air. For purposes of the heat transfer calculations, a *geometrically* correct rendering of the zone surfaces is not required. The surfaces do not even have to be connected. As long as the program knows to which thermal zone (mass of air) each surface transfers heat, it will calculate all heat balances correctly. For example, all heat storage surfaces of the same construction within a zone may be defined as a single rectangular surface. The size of this *equivalent* surface will equal the sum of all the areas of all the heat storage surfaces in the zone. A few simple rules will further explain what we mean by *equivalent* surfaces and where these surfaces should be used.

1. *Define all roofs and floors as rectangles regardless of the shape of the zone.* Each zone should have one rectangular roof and one rectangular floor of a given construction.

2. *Define all heat storage surfaces of the same construction within a zone as a single surface*. The size of the single surface is obtained by summing the individual surface areas exposed to the zone. Thus, if a partition is completely within a zone (both sides of the partition are exposed to the zone), the area of each side must be added to the area of the equivalent surface. On the other hand, if the partition separates two zones, the area of only one side should be added to the equivalent surface.

3. *Combine all windows on a given exterior surface into a single window*. Usually each exterior surface should have only one window of each type. Overhangs or other shading devices may require that more windows be specified or combined together. By using the WindowGlass construction for your glass door, they will be correctly modeled in EnergyPlus with sunlight transferring into the zone.

The following figure shows the surfaces and subsurfaces required for a one-zone model, i.e., the Fort Monmouth education center. Since there were two types of partitions in the building, two heat storage surfaces ("internal mass") of different constructions were defined.

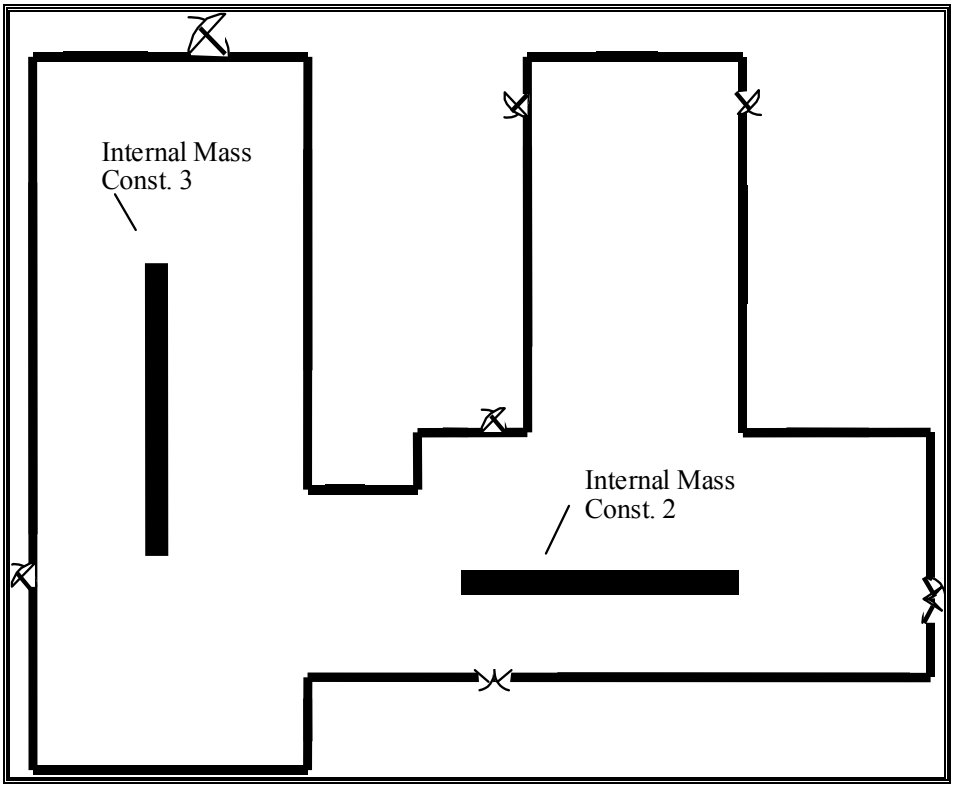

Figure 9. Simplifications Using Equivalent Surfaces

Step3.3. Specify construction elements

BLAST, DOE-2 and other programs often have "libraries" of constructions, schedules, and other aspects of simulating the building. Since EnergyPlus has been designed as an engine, we are relying on EnergyPlus Interface Developers to embody these kinds of databases within their user-friendly programs. In many cases, the EnergyPlus team will be helping this development by supplying Reference Data Sets (RDS), such as equivalent IDF constructions and materials for all the BLAST or DOE-2 libraries. Even then, the EnergyPlus Input Data File will likely be a "complete" file – all elements needed to run the simulation will be incorporated into the single Input Data File (IDF). The following table uses names of materials that are contained in the BLAST libraries. Like BLAST, EnergyPlus requires that materials are listed from the outside to the inside of the zone. The following table shows the construction elements for the Education Center.

Number	Type (1)	Name (2)	Abbr. (3)	Material (4)
1	Wall	const1	A2	4 IN DENSE FACE BRICK
			C8	8 IN HW CONCRETE BLOCK
			IN3	MINERAL FIBER FIBROUS 6 IN
			PL4	GYPSUM LWA 5 / 8 IN
2	Partition	const2	PL4	GYPSUM LWA 5 / 8 IN
			A2	4 IN DENSE FACE BRICK
			C8	8 IN HW CONCRETE BLOCK
			PL4	GYPSUM LWA 5 / 8 IN

3	Partition	const3	PL4	GYPSUM LWA 5 / 8 IN
			B1	AIRSPACE RESISTANCE
			PL4	GYPSUM LWA 5 / 8 IN
4	Wall	const4	PL4	GYPSUM LWA 5 / 8 IN
			C8	8 IN HW CONCRETE BLOCK
			A2	4 IN DENSE FACE BRICK
			PL4	GYPSUM LWA 5 / 8 IN
5	Roof	const5	RF4	BUILT UP ROOFING 3 / 8 IN
			IN71	EXPANDED EXT POLYSTYRENE R12 2 IN
			E4	CEILING AIRSPACE
			BB17	ACOUSTIC TILE 3 / 4 IN
6	Floor	const6	DIRT 12 IN	DIRT 12 IN
			CO17	CONCRETE DRIED SAND AND GRAVEL 4 IN
			FF5	FINISH FLOORING TILE 1 / 16 IN

Table 4. Building Elements

Notes:

(1) The surface type is either a wall, floor, roof, window or door.

(2) User supplies name for the element.

(3) Abbreviate as found in the BLAST Standard libraries.

(4) Material's full name is as found in the BLAST Standard libraries.

Step 3.4. Compile surface and subsurface information.

Building information:

1. *Building North Axis:* This syntax simplifies building geometry specification by designating one wall of the building as the building's north pointing axis. The building model North axis is measured from true (compass) North. Surface facing angles (see surface information below) are then specified relative to the building north axis. The *North Axis* entry in the *BLAST User Reference* illustrates specification of the building north axis.

Zone information:

1. *Wall height*: This is entered once. All walls are assumed to be the same height. If the height for a given wall differs from the specified height, the wall length should be adjusted by the user to give the correct equivalent area. In certain conditions this may not be possible and you will need to resize each wall accordingly.

Surface information:

1. *Base Surface Type:* Heat Transfer/Heat Storage Surfaces may be of the following types: wall, floor, roof, internal mass, or subsurface

2. *Construction:* The type of construction of the surface (see previous table).

Subsurface information:

1. *Subsurfaces* are Windows, Doors or GlassDoors

2. *Area:* Area of the subsurface.

3. *Reveal:* For windows only, the distance it is inset from the outside surface of a wall.

Consider the single zone model. The following figure is a schematic representation of a one zone representation. The figure shows the location of the "zone origin", the length of all

"base" surfaces and the areas of all "subsurfaces" (windows and doors). The surfaces are numbered counter-clockwise around the zone beginning at the zone origin as shown the table below, the table of required zone information compiled by the user. A few simple conventions should be followed to facilitate the construction of zone information tables:

1. Number all surfaces in order counter-clockwise around the zone.
2. Keep the subsurfaces with the base surface on which they are located.
3. Specify *lengths* for base surfaces and areas for subsurfaces and internal mass.
4. Specify the roof and floor as rectangles of the correct size.

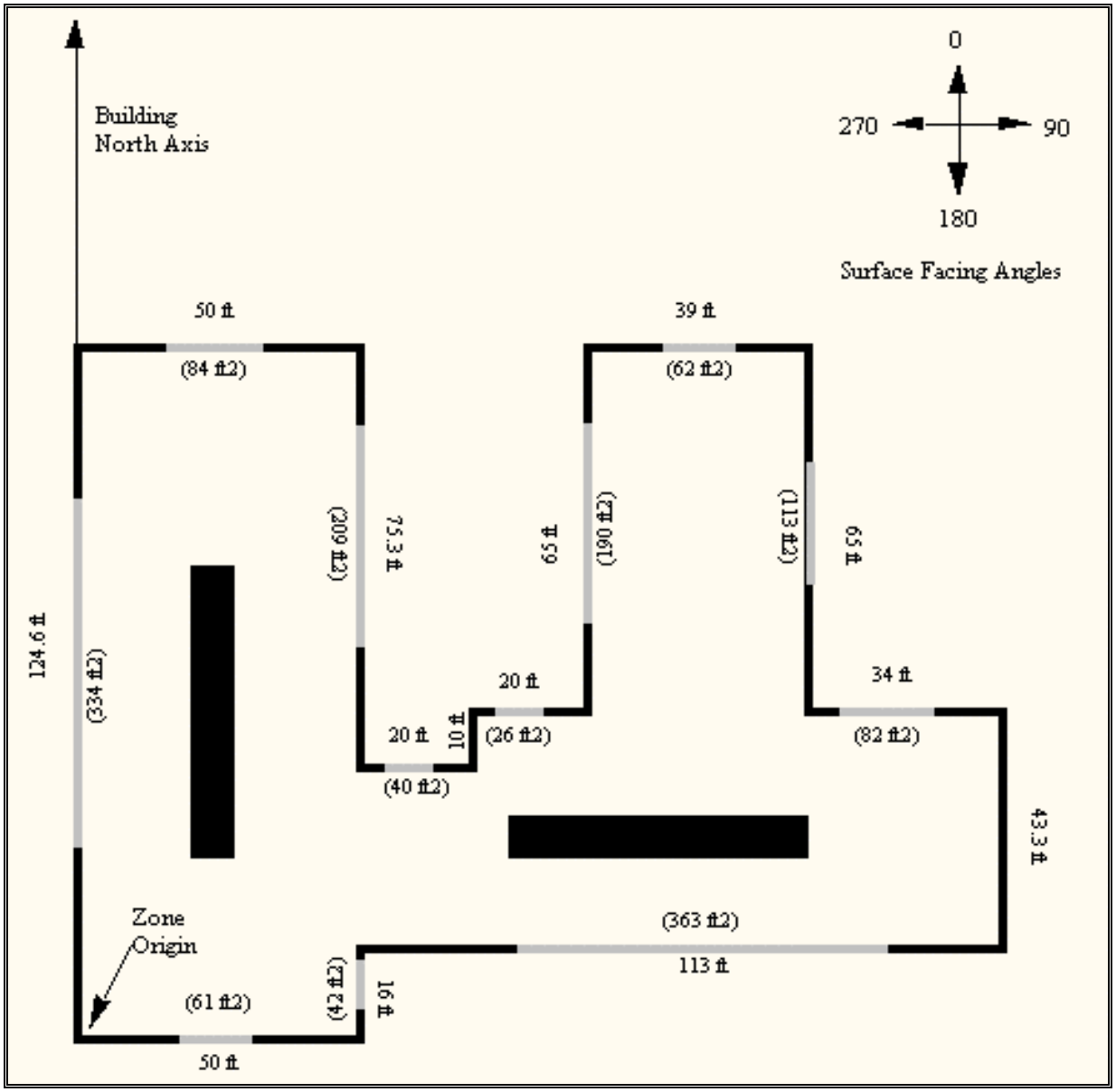

Figure 10. Schematic of One Zone Model with Surface Information.

Zone 1

Surface	type	construction	length	area
1	exterior wall	const1	50	

2	window	dpw		61
3	exterior wall	const1	16	
4	window	dpw		42
5	exterior wall	const1	113	
6	window	dpw		363
7	exterior wall	const1	43.33	
8	window	dpw		101
9	exterior wall	const1	34	
10	window	dpw		82
11	exterior wall	const1	65	
12	window	dpw		113
13	exterior wall	const1	39	
14	window	dpw		62
15	exterior wall	const1	65	
16	window	dpw		190
17	exterior wall	const1	20	
18	window	dpw		26
19	exterior wall	const1	10	
20	exterior wall	const1	20	
21	window	dpw		40
22	exterior wall	const1	75.33	
23	window	dpw		209
24	exterior wall	const1	50	
25	window	dpw		84
26	exterior wall	const1	124.66	
27	window	dpw		334
28	roof	const5	116 X 116	13450
29	floor	const6	116 X 116	13450
30	internal mass	const2		10300
31	internal mass	const3		18920

Table 5. Compilation of Surface Information for the One Zone Model

The column headings in the previous table have the following meanings:

Type: A shortened notation for the surface type in EnergyPlus to differentiate between heat storage surfaces and various types of heat transfer surfaces.

Construction: A name for the surface construction types.

Length: The length of base surfaces. An equivalent size of roofs and floors is also included.

Area: The area of subsurfaces.

Step 4: **Compile Internal Space Gain Data**

People, lights, equipment, outside air infiltration and ventilation all constitute "internal gains" for the thermal zone. These gains are described to EnergyPlus as a *design or peak* level with a *schedule* that specifies a fraction of the peak for each hour. The peak level is

calculated by the user. Table 6. Internal Gain Data shows the internal loads for a single zone model of Ft. Monmouth and the schedule named to specify the hourly load.

Table 6. Internal Gain Data

Zone	Gain Type	Size	Schedule
1	People	205	Office occupancy
	Lights	90	Office lighting
	Infiltration	1570	Constant

The column headings in the table have the following meanings:

Gain Type: The code used to differentiate between various types of internal gains.

Size: The peak load. This is the actual size of the load for every hour that the schedule specifies "100%".

Schedule: The hourly schedule which specifies the percentage of peak load for each hour of the day.

Creating Input Files

This version of EnergyPlus has several options for the user to create input files. One of the goals of EnergyPlus was to make a simple, readable input file for the program. More of this background and explanation is contained in the Interface Developer's Guide. Key to reading the input file is understanding the Input Data Dictionary. Recently, we have set some conventions for commenting the IDD so that the units, minimum, maximum, and other information. This changes some of the information that is shown in the Interface Developer's guide. The most current information is in the actual Energy+.idd delivered with the program.

Four methods are available to create input files:

1) IDFEditor – this is a very simple, "intelligent" editor that reads the IDD and IDFs and allows creation/revision of IDF files. It can be run from a shortcut in the main EnergyPlus directory (created as part of the install) or directly from EP-Launch.
2) BLAST Translator – if you already have BLAST and/or BLAST input files, this program will produce the bulk of a translation to EnergyPlus for you. It generates a complete IDF file but does not include specifics for Systems or Plants. (It does include the System and Plant schedules that were in the BLAST deck). Many of the sample files included with the install started out as BLAST input files.
3) DOE-2 Translator – if you already have DOE-2.1e input files, this program will produce the bulk of a translation to EnergyPlus for you. It generates a IMF (input macro file) that must be run through the EnergyPlus Macro (EPMacro) program before it can be used by EnergyPlus.
4) Hand editing – for simple changes to an existing file (such as one of the sample files), you can hand edit a file using your knowledge of the IDD, comments in the IDF file, and a text editor such as NOTEPAD™. For creating HVAC simulations – the HVAC templates are also available.

IDD Conventions

The following is a basic description of the structure of the IDD (it's actually taken directly from the IDD file). As noted wihin, ! signifies a comment character as does the \. \ has also been adopted as a convention for including more specific comments about each field in an object. These have been used with success in the IDFEditor and it is hoped the flexibility will provide other interface developers with useful information. Not all fields are filled in as of yet in the existing IDD – that will come over time.

```
! ***************************************************************************
!
! Object Description
! ------------------
! To define an object (a record with data), develop a key word that is unique
! Each data item to the object can be A (Alphanumeric string) or N (numeric)
! Number each A and N.  This will show how the data items will be put into the
! arrays that are passed to the Input Processor "Get" (GetObjectItem) routines.
! All alpha fields are limited to 40 characters.  Numeric fields should be
! valid numerics (can include such as 1.0E+05) and are placed into single
! precision variables.
!
! Object Documentation
! --------------------
! In addition, the following special comments appear one per line and
! are followed by a value.  Comments may apply to a field, and object
! or a group of objects.
!
! Field-level comments:
!
```

```
!  \field            Name of field
!                      (should be succinct and readable, blanks are encouraged)
!
!  \note             Note describing the field and its valid values
!
!  \required-field   To flag fields which may not be left blank
!                      (this comment has no "value")
!
!  \units            Units (must be from EnergyPlus standard units list)
!                    EnergyPlus units are standard SI units
!
!  \ip-units         IP-Units (for use by input processors with IP units)
!
!  \minimum          Minimum that includes the following value
!
!  \minimum>         Minimum that must be > than the following value
!
!  \maximum          Maximum that includes the following value
!
!  \maximum<         Maximum that must be < than the following value
!
!  \default          Default for the field (if N/A then omit entire line)
!
!  \type             Type of data for the field -
!                      integer
!                      real
!                      alpha       (arbitrary string),
!                      choice      (alpha with specific list of choices, see
\key)
!                      object-list (link to a list of objects defined elsewhere,
!                                   see \object-list and \reference)
!
!  \key              Possible value for "\type choice" (blanks are significant)
!                      use multiple \key lines to indicate all valid choices
!
!  \object-list      Name of a list of object names that are valid entries
!                      for this field (used with "\reference")
!                      see ZONE and Surface:HeatTransfer objects below for
examples
!
!  \reference        Name of a list of object names to which this object belongs
!                      used with "\type object-list" and with "\object-list"
!                      see ZONE and Surface:HeatTransfer objects below for
examples:
!
!                         ZONE,
!                           A1 , \field Zone Name
!                                \type alpha
!                                \reference ZoneNames
!                         Surface:HeatTransfer,
!                           A4 , \field InsideFaceEnvironment
!                                \note Zone the surface is a part of
!                                \type object-list
!                                \object-list ZoneNames
!
!                      For each zone, the field "Zone Name" may be referenced
!                      by other objects, such as Surface:HeatTransfer, so it is
commented
!                      with "\reference ZoneNames"
!                      Fields that reference a zone name, such as a
Surface:HeatTransfer's
!                      "InsideFaceEnvironment", are commented as
!                      "\type object-list" and "\object-list ZoneNames"
```

```
!
! Object-level comments:
!
!  \memo             Memo describing the object
!
!  \unique-object    To flag objects which should appear only once in an idf
!                      (this comment has no "value")
!
!  \required-object To flag objects which are required in every idf
!                      (this comment has no "value")
!
! Group-level comments:
!
!  \group           Name for a group of related objects
!
!
! Notes on comments
! -----------------
!
! 1.  If a particular comment is not applicable (such as units, or default)
! then simply omit the comment rather than indicating N/A.
!
! 2.  Memos and notes should be brief (recommend 5 lines or less per block).
! More extensive explanations are expected to be in the user documentation
!
! ****************************************************************************
```

Standard EnergyPlus Units

EnergyPlus has adopted the standard SI units for input and output.

ASCII with no spaces is used for abbreviations. Note that exponents appear without any indication of exponentiation: i.e., kg/m3 not kg/m^3 or kg/m**3. Also note the use of dashes. We have W/m2-K not W/m2*K or W/(m2*K).

At the end we note the “problem” variables – the inputs that have non-standard units. Inputs using these units will have to be changed and the code checked to see how the quantities are used internally.

Table 7. Standard EnergyPlus Units

Quantity	unit	abbreviation
angular degrees	degree	deg
Length	meter	m
Area	square meter	m2
Volume	cubic meter	m3
Time	seconds	s
frequency	Hertz	Hz
temperature	Celsius	C
absolute temperature	Kelvin	K
temperature difference	Kelvin	delK
Speed	meters per second	m/s
energy (or work)	Joules	J
Power	Watts	W
Mass	kilograms	kg

Force	Newton	N
mass flow	kilograms per second	kg/s
volume flow	cubic meters per second	m3/s
Pressure	Pascals	Pa
pressure difference	Pascals	delPa
specific enthalpy	Joules per kilogram	J/kg
Density	kilograms per cubic meter	kg/m3
heat flux	watts per square meter	W/m2
specific heat	-------	J/kg-K
conductivity	-------	W/m-K
Diffusivity	-------	m2/s
heat transfer coefficient	-------	W/m2-K
R-value	-------	m2-K/W
heating or cooling capacity	Watts	W
electric potential	volts	V
electric current	Amperes	A
illuminace	lux	lx
luminous flux	lumen	lm
luminous intensity	candelas	cd
luminance	candelas per square meter	cd/m2
vapor diffusivity	m2/s	
Viscosity	-------	kg/m-s
Porosity	-------	m3/m3
thermal gradient coeff for moisture capacity	-------	kg/kg-K
isothermal moisture capacity	-------	m3/kg

IDFEditor

IDF Editor is an optional component of the EnergyPlus installation. For users who want a simple way of creating or editing EnergyPlus input data files (IDF), IDF Editor provides this service. There is no checking of data entered by the user to ensure that values are valid or within range, although for each field the range of allowable values is displayed. For instructions and rules that must be followed when creating an IDF file the user should refer to the *Input/Output Reference* document.

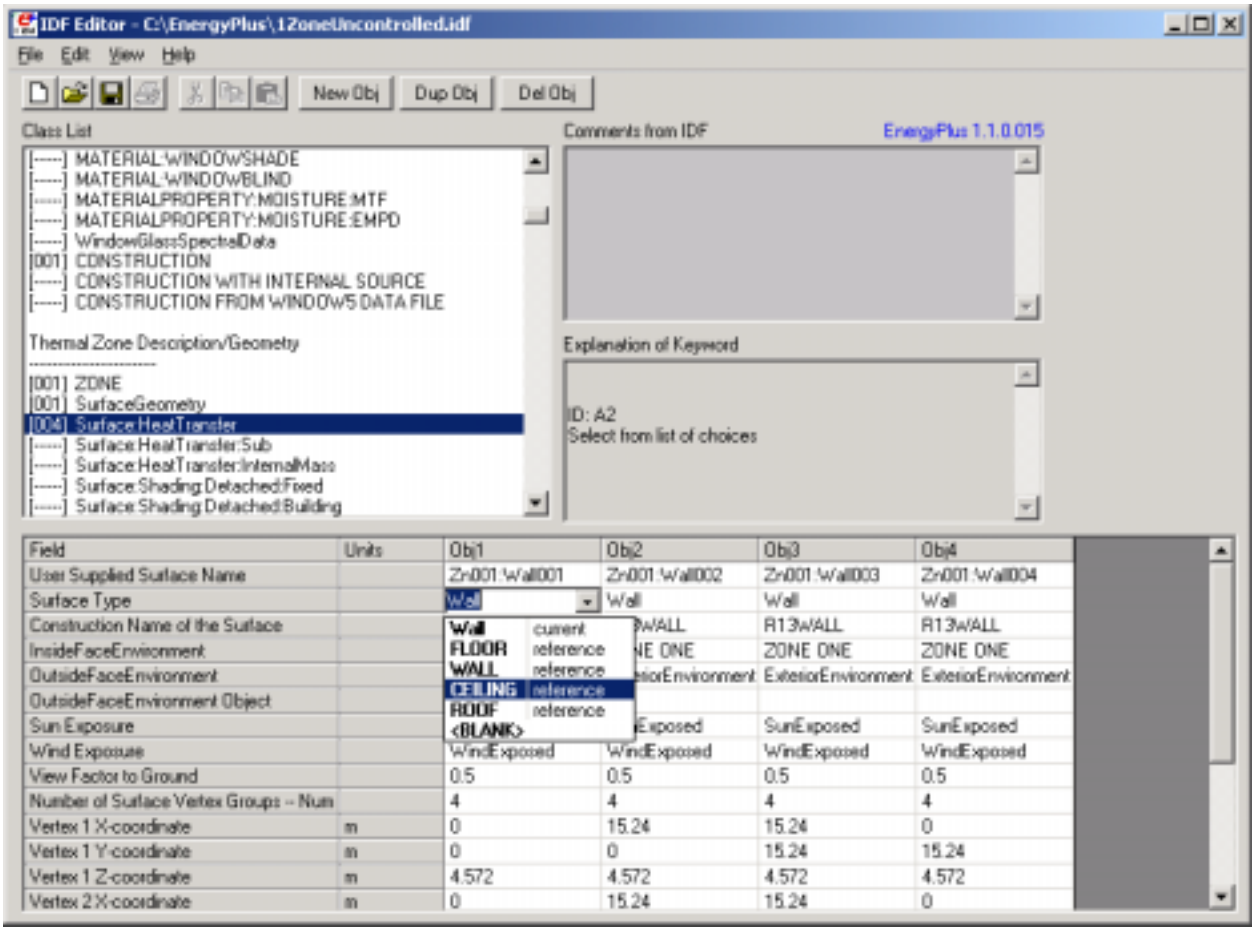

Figure 11. IDF Editor Screen

Start IDF Editor

IDF Editor should be located in the EnergyPlus\PreProcessor\IDFEditor directory where you installed EnergyPlus. By double clicking on the IDF Editor icon you will get a screen similar to the one shown above. This is the only screen that the IDF Editor uses. IDF Editor works in conjunction with the current EnergyPlus Input Data Directory (IDD) file that resides in the directory where EnergyPlus is installed. Another way to start the IDF Editor is from EP-Launch.

Creating or Selecting an Input Data File

Creating a new input data file or selecting an existing input data file can be accomplished either through use of the File menu on the menu bar at the top of the screen or through use of the New File icon button or Open File icon button on the tool bar. Only one file can be open at any time.

Class List and Objects

The classes that can be used to make up an IDF file have been organized into groups as shown in the 'Class List' portion of the screen. A class is made up of a group of objects. Select a class from the list by clicking on and highlighting the class. The field to the left of the selected class in the 'Class List' will either contain [------] to indicate that this class has no objects in the IDF file or it will contain a number like [003] to indicate the number of times the object currently appears in the IDF file. For example, for the SURFACE class selected in the screen above under the ZONE group, there are 8 objects in the IDF file. The details for these

8 objects or any new object that is defined are displayed in columns within the grid. Each object is made up of fields and can be used to further define the object. Any units attached to each field are shown in the second column. You may need to scroll down the 'field' list or maximize the application to see all of the fields. Likewise, you may need to scroll to the right of the main grid to see other objects.

Changing Values

By clicking and highlighting a value within an object, several things happen:

1) Any user comments from the IDF file will be displayed in the 'Comments from IDF' portion of the screen

2) Any notes contained in the IDD for this input field will be displayed in the 'Explanation of Keyword' portion of the screen

3) The value can be edited. Depending on the field, a drop down list may display the default value, maximum and minimum, or other keywords that can be used with the field.

4) Numeric fields which can be autosized will include "autosize" as a selection in the drop down list.

Working with Objects

To delete an object, first click on any value for the object and then click on the "Del Obj" button. To add a new object, click on the "New Obj" button and a new object column with fields set to blanks or zeros will be added to the far right of the grid. The "Dup Obj" button is similar to "New Obj", but copies the values of the fields of the currently selected object. (Also see the Edit menu to perform these functions.)

File Menu

The File menu can be used for creating or selecting input files just like the buttons on the IDF Editor screen (see the *Creating or Selecting an Input File* section above). In addition, the File menu is used to save a file or exit the IDF Editor.

Edit Menu

The Edit Menu offers options to create a new object, duplicate an object, and delete an object. These same operations can also be accomplished by using the 'New Obj', 'Dup Obj' and 'Del Obj' buttons (see the *Working with Objects* section above). In addition, the "Next Row after Enter" option can be toggled. When this option is on, the selection moves down one row after pressing Enter.

View Menu

The View menu offers options for units and column widths. The Narrow/Medium/Wide Column options set the standard column width for items in the object grid. Individual columns can also be resized by dragging the column separator. The displayed value is rounded and/or expressed in scientific notation to fit within the column width.

EnergyPlus input files must always be in SI units, and the IDF Editor always opens in SI units mode. Selecting "Inch-Pound" (IP) units in the View menu displays and edits values in IP units. IP units are only partially supported:

1) Fields with no units specified are not converted. This includes all **schedules** and **fluid properties**. If a field has no units displayed in the units column of the object grid, then no conversion will be made.

> Note that thermostat setpoints, people activity levels, and other dimensioned schedules must always be entered in SI units, even when the IDF Editor is in Inch-Pound mode. These objects which have no units field will not be converted.

2) The IP unit will be displayed in the units column of the object grid. Some SI units convert to multiple IP units. For example, W becomes Btu/hr for heating and cooling capacity but remains as W for lighting and electrical equipment.
3) All conversion factors used in the IDF editor are documented in a block of comments near the top of the Energy+.IDD file.

Help Menu

The Help menu offers options to open the EnergyPlus documentation files.

Caveats

Remember to save any changes made before you create or edit another input file.

No “Run EnergyPlus” button is available. Save your IDF file and use EP-Launch to execute an EnergyPlus run.

You cannot edit comments in the ‘Comments from IDF’ section of the screen.

All engineering units are in SI. IP units have not been implemented yet.

The use of point "." or comma "," as the decimal symbol is controlled by the windows system settings. This setting is found in the Control Panel, Regional Options, Number tab, Decimal Symbol field. IDF Editor will use the current decimal symbol to signify the start of the fractional portion of the number and will ignore other symbols. The idf file is always written using point "." as the decimal symbol.

Bugs

Please report any bugs to the energyplus-support@gard.com address so that we can fix them prior to the next release.

BLASTTranslator

The BLAST Translator will produce an IDF file from an existing BLAST Input File (usually called <something>.bin. For anyone that is unfamiliar, BLAST stands for the Building Loads Analysis and Systems Thermodynamics computer program.

A special batch file (**RunXLate.bat**) can be used to run the translation program. Similar to running EnergyPlus (see above), you run this batch file:

RunXLate <blastinputfile>

Where <blastinputfile> is the part of the file name without extension. The program produces a .idf file of the same name.

The BLASTTranslator uses an “Energy+.ini” file for some parameters. For example:

```
[program]
dir=
ver=
surf=group

[weather]
dir=
```

The above BLASTTranslator ini file sets the “version” of EnergyPlus to the current version and has the “surf” parameter set to “group”. This BLASTTranslator run will produce an EnergyPlus input file for the current release version format and will name surfaces by Zone and Class (e.g. ZN001:Wall001). The alternative “Consecutive” will number surfaces in sequence.

DOE2Translator

The DOE2Translator program creates an EnergyPlus input file from a DOE-2.1E input file. The translation is not intended to be complete but simply an aid to help you move your library of DOE-2.1E input files into EnergyPlus. You should look over the resulting EnergyPlus input file, review the documentation of EnergyPlus, and make any necessary edits to the translated file.

To use the DOE2Translator program, at the DOS prompt (or the command prompt for Windows NT/2000 systems), go to the directory that the DOE2Translator is located. That directory is likely to be:

c:\EnergyPlus\PreProcess\DOE2Translator

In this directory there should be several files:

DOE2Translator.exe - the program

D2E_macr.txt - a support file that contains the header of the translated file

D2comkey.dat - a support file that contains a list of DOE-2 keywords

Samp4n.inp - sample DOE-2.1E input file

Samp4n.imf - the EnergyPlus macro input file resulting from the sample

To use the DOE2Translator simply type

DOE2Translator <file>

Where you substitute the file you want to translate for <file> without a file extension. The ".inp" file extension is assumed. For example, if you want to translate one of the sample DOE-2.1E input files you would type:

DOE2Translator samp1b

The <file> can also have a full path, but it should not have an extension.

Several files get created when you run the DOE2Translator program. In the same directory as the DOE-2.1E input file, a new file with the same name and the file extension of ".imf" contains the EnergyPlus translation.

This is an EnergyPlus macro file and the macro processor EPMacro needs to be used. The DOE2Tranlator uses many macros so using EPMacro is essensial. EP-Launch automatically runs EP-Macro when an ".imf" file is selected. In the translated file, comments shown with a tilde "~" are messages from the DOE2Translator program that may be helpful in understanding the limits of the translation.

The D2EP.log file contains a detailed log of the translation process. The D2E_TEMP.txt file contains an intermediate version of the log file. Both of these files are created in the same directory as the DOE2Translator program and can usually be deleted.

In this version of the DOE2Translator program, translation is limited to the following DOE-2 commands, which represent the majority of the building envelope and internal gains:

```
SPACE (except SHAPE=BOX)
SPACE-CONDITIONS
DAY-SCHEDULE (except use of HOURS and VALUES keywords)
WEEK-SCHEDULE (except use of DAYS and DAY-SCHEDULE keywords)
SCHEDULE (except use of WEEK-SCHEDULE keyword)
MATERIAL
LAYERS
CONSTRUCTION
EXTERIOR-WALL, ROOF (except polygon)
INTERIOR-WALL
FIXED-SHADE
WINDOW
DOOR
RUN-PERIOD
DESIGN-DAY
```

```
    LIKE
    SET-DEFAULT
```

IFCtoIDF

The IFCtoIDF utility extracts the geometric representation of building, space, and envelope surface and opening object instances contained within an Industry Foundation Classes (IFC) data file – then creates an EnergyPlus Input Data File (IDF) containing these objects mapped to the EnergyPlus Input Data Dictionary (IDD). The IFCtoIDF utility deals only with the geometry of building spaces and surfaces, without addressing the material characteristics of these building elements. It is anticipated that this utility will be enhanced over time to extract and map other information contained in an IFC data file that is relevant to performing an EnergyPlus simulation.

The utility has been developed as a Window™ dynamic link library (DLL) that reads an IFC data file, maps the geometry of relevant object instances to EnergyPlus objects, and writes an EnergyPlus IDF. This is not an interactive process in which the user can intervene to make changes or additions to the resulting IDF. A simple host graphical user interface (GUI) application has also been developed that allows an end user to specify the IFC data file to be read, and the IDF to create. The GUI and the DLL run on the Microsoft Windows 95/98/NT platforms. The utility is built upon, and requires the installation of, the BSPro COM-Server software package developed by Olof Granlund Oy of Finland (see www.bspro.net and Section 5.2 in the IFCtoIDF user manual). The BSPro COM-Server software would be necessary for you to extract the geometry information from an IFC compliant tool. If you choose the IFCtoIDF utility, the BSPro COM-Server is installed as well.

The IFCtoIDF utility is primarily intended as an example of the capability of developing interoperability between commonly available software (e.g., CAD tools) and EnergyPlus, using the IFC data model. It is anticipated that as more software tools become IFC-compliant, the capabilities demonstrated in the IFCtoIDF utility will be incorporated into a variety of other software, including specialized user interfaces for EnergyPlus.

Text Editor

Text editors, such as NOTEPAD, can also be used to edit an IDF file. Do not use tabs or other special characters in the IDF file as EnergyPlus will not convert these properly.

HVAC Templates

The HVAC templates are described in detail in the Other Information Document.

WeatherData

As mentioned in the "Running EnergyPlus" section, EnergyPlus uses the new E/E+ weather data.

The new E/E+ format is very flexible (as well as being ASCII and somewhat readable). In addition to the usual weather data (temperatures, solar radiation data), the format embodies other information from the location and weather data (e.g. design conditions, calculated ground temperatures, typical and extreme weather periods). The EPW (weather data format) is described in the Input Output Reference Document and Auxiliary Programs Document. Other details including statistical reports, backgrounds on data sources and formats, use of the Weather Converter program (used both for processing data and reporting) are also provided in the Auxiliary Programs document.

The web site for EnergyPlus (http://www.energyplus.gov) provides downloadable weather data for many sites throughout the world from several different formats. In addition, we are amenable to posting more weather data from users.

In summary, the web site contains data sets of the following formats:

Table 8. Formats of weather data on the web site

Data Format	Number of Sites
IWEC	227
CWEC	55
TMY	39
TMY2	239

And, by locations, the following demographics are observed:

Table 9. General Locations of Weather Sites on web page

Location	Number of sites
International	233
Canada	55
US	275

The US sites include the 16 sites for California Climates used in compliance for California Energy Codes.

PostProcessing

ReadVarsESO

The postprocessing program that is installed (ReadVarESO) is a simple program that reads the **eplusout.eso** or **eplusout.mtr** output files and produces a simple comma separated format that is easily accepted by a spreadsheet program such as Excel™.

```
ReadVarsESO.exe eplusout.inp
```

If you do not specify a file of input commands, ReadVarsESO will produce its output file using all the variables available on the eplusout.eso file.

PostProcessing Files details

The following files are used or produced by this postprocessor.

eplusout.inp

This is an ascii (text) file read by RunReadESO.bat. The first line contains the name of the ESO file that the postprocessor will read. The second line contains the name of the Excel compatible output file that it will write. The subsequent lines specify which variables' data will be extracted from the ESO file. The variables may be specified by unique number or by name, one variable per line. The file is terminated with a line containing "0". An example of eplusout.inp is:

```
eplusout.eso
eplusout.csv
Outdoor Dry Bulb
PSI FOYER,Mean Air Temperature
DORM ROOMS AND COMMON AREAS,Mean Air Temperature
LEFT FORK,Mean Air Temperature
MIDDLE FORK,Mean Air Temperature
RIGHT FORK,Mean Air Temperature
PSI FOYER,Zone-Total Electric Consumption
DORM ROOMS AND COMMON AREAS,Zone-Total Electric Consumption
LEFT FORK,Zone-Total Electric Consumption
MIDDLE FORK,Zone-Total Electric Consumption
RIGHT FORK,Zone-Total Electric Consumption
0
```

eplusout.eso

This is the standard output file from EnergyPlus. The above example is meant to be applied to the ESO from the GeometryTest test input. The "data dictionary" section of the GeometryTest ESO file looks like:

```
6,2,Environment,Outdoor Dry Bulb [C] !Hourly
21,2,PSI FOYER,Mean Air Temperature[C] !Hourly
22,2,PSI FOYER,Zone-Total Latent Gain[J] !Hourly
26,2,PSI FOYER,Zone-Total Electric Consumption[J] !Hourly
27,2,DORM ROOMS AND COMMON AREAS,Mean Air Temperature[C] !Hourly
28,2,DORM ROOMS AND COMMON AREAS,Zone-Total Latent Gain[J] !Hourly
32,2,DORM ROOMS AND COMMON AREAS,Zone-Total Electric Consumption[J] !Hourly
33,2,LEFT FORK,Mean Air Temperature[C] !Hourly
34,2,LEFT FORK,Zone-Total Latent Gain[J] !Hourly
38,2,LEFT FORK,Zone-Total Electric Consumption[J] !Hourly
```

```
39,2,MIDDLE FORK,Mean Air Temperature[C] !Hourly
40,2,MIDDLE FORK,Zone-Total Latent Gain[J] !Hourly
44,2,MIDDLE FORK,Zone-Total Electric Consumption[J] !Hourly
45,2,RIGHT FORK,Mean Air Temperature[C] !Hourly
46,2,RIGHT FORK,Zone-Total Latent Gain[J] !Hourly
50,2,RIGHT FORK,Zone-Total Electric Consumption[J] !Hourly
76,2,PSI FOYER,Lights-Return Air Heat Gain[J] !Hourly
77,2,PSI FOYER,Lights-Radiant Heat Gain[J] !Hourly
80,2,PSI FOYER,Lights-Total Heat Gain[J] !Hourly
81,2,PSI FOYER,Lights-Electric Consumption[J] !Hourly
82,2,DORM ROOMS AND COMMON AREAS,Lights-Return Air Heat Gain[J] !Hourly
83,2,DORM ROOMS AND COMMON AREAS,Lights-Radiant Heat Gain[J] !Hourly
86,2,DORM ROOMS AND COMMON AREAS,Lights-Total Heat Gain[J] !Hourly
87,2,DORM ROOMS AND COMMON AREAS,Lights-Electric Consumption[J] !Hourly
88,2,LEFT FORK,Lights-Return Air Heat Gain[J] !Hourly
89,2,LEFT FORK,Lights-Radiant Heat Gain[J] !Hourly
92,2,LEFT FORK,Lights-Total Heat Gain[J] !Hourly
93,2,LEFT FORK,Lights-Electric Consumption[J] !Hourly
94,2,MIDDLE FORK,Lights-Return Air Heat Gain[J] !Hourly
95,2,MIDDLE FORK,Lights-Radiant Heat Gain[J] !Hourly
98,2,MIDDLE FORK,Lights-Total Heat Gain[J] !Hourly
99,2,MIDDLE FORK,Lights-Electric Consumption[J] !Hourly
100,2,RIGHT FORK,Lights-Return Air Heat Gain[J] !Hourly
101,2,RIGHT FORK,Lights-Radiant Heat Gain[J] !Hourly
104,2,RIGHT FORK,Lights-Total Heat Gain[J] !Hourly
105,2,RIGHT FORK,Lights-Electric Consumption[J] !Hourly
End of Data Dictionary
```

Each variable is uniquely specified by a number (6, 18, 19, etc.) or by the combination of object name (in this case a zone) and variable name. The eplusout.inp file could alternatively been specified as:

```
eplusout.eso
eplusout.csv
6
21
27
33
39
45
26
32
38
44
50
0
```

eplusout.csv

This is the *comma delimited* file output from the postprocessor. It contains data for selected variables from an ESO file. It is a comma separated text file suitable for input to a speadsheet such as Excel.

**.rvi*

Files with the *.rvi* extension are just *eplusout.inp* files for a specific simulation input. For instance, GeometryTest.rvi is the eplusout.inp file for the simulation described by GeometryTest.idf. Since each .rvi file makes sense only relative to a specific input, the .rvi files are generally stored in the same directory as the input (IDF) files.

DataFiles

Many example files are optionally installed during installation (Sample Files option). Each sample input file should contain comments about its purpose at the start of the file. For several options (including autosizing), the outputs are also installed.

HVAC Templates are installed with the sample files option. More information about the use of HVAC templates is contained in the Auxiliary Programs document.

Energy Meters

Meters will provide one way for EnergyPlus to report energy use in a form that is pallatable to the users. The primary implemented method for output gives very fine detail (down to the variable level) for results from EnergyPlus. However, to get the required energy use, there may be several variables that need to be polled and accumulated. The meter implementation for EnergyPlus will accomplish this reporting.

Meters can be used to typify energy use by type and by component. The following diagrams and tables will illustrate how the meters have been incorporated into EnergyPlus.

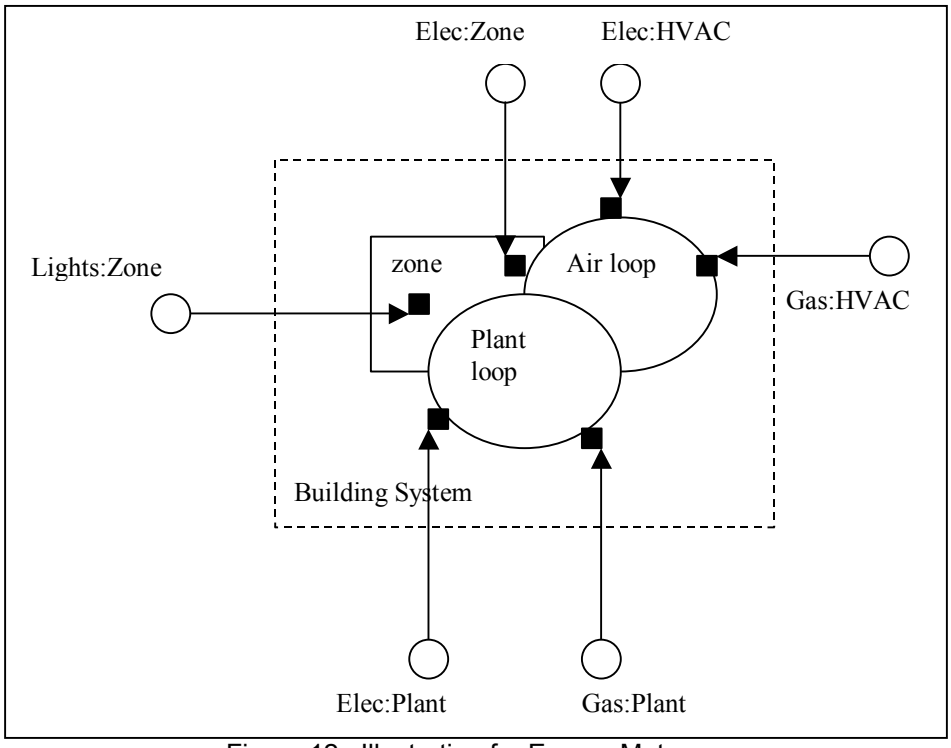

Figure 12. Illustration for Energy Meters

As shown in the figure above, energy use for the facility is grouped according to fuel type (see Table 11. Table of Metered Fuel Types), by meter type (see Table 10. Overall Meter Types) and by end use category type (see Table 12. End Use Category Types). The developers place special calls which allow for the appropriate grouping and thence the accumulation of values.

For each simulation, the meter names will be shown in the Report Variable Dictionary file:

```
Var Type,Var Report Type,Variable Name [Units]
Zone,Meter,Electricity:Facility [J]
Zone,Meter,Electricity:Building [J]
Zone,Meter,Electricity:Zone:RESISTIVE ZONE [J]
Zone,Meter,Electricity:Zone:EAST ZONE [J]
Zone,Meter,Electricity:Zone:NORTH ZONE [J]
Zone,Meter,GeneralLights:Electricity [J]
Zone,Meter,GeneralLights:Electricity:Zone:RESISTIVE ZONE [J]
Zone,Meter,GeneralLights:Electricity:Zone:EAST ZONE [J]
Zone,Meter,GeneralLights:Electricity:Zone:NORTH ZONE [J]
```

The VarType=Zone field designates that each meter will be available at the Zone Time Step resolution – that is, whatever is specified by the TimeSteps in Hour user input.

The Var Report Type=Meter field shows that these are “meters”.

The Variable Name (really the Meter Name) shows the name as well as the units for reporting. Units illustrate the units for these meters. We are currently allowing two possibilities for "water" units (both m^3 for water that is "consumed" on the property as well as joules for water that might be purchased for energy use or other hot/cold energy water usage).

Table 10. Overall Meter Types

Meters
Facility
Building
Zone
System
Plant

You can report on meters at the zone time step, but you can also have the reporting at other frequencies (e.g. hourly, daily, monthly, overall simulation). For the more general resolutions, the values of the meter are shown along with minimum and maximum values. For example:

```
26,Electricity:Facility [J]  ! RunPeriod
301,Electricity:Building [J]  ! RunPeriod
306,Electricity:Zone:RESISTIVE ZONE [J]  ! RunPeriod
311,Electricity:Zone:EAST ZONE [J]  ! RunPeriod
316,Electricity:Zone:NORTH ZONE [J]  ! RunPeriod
42,ExteriorEquipment:Gas [J]  ! RunPeriod
53,ExteriorEquipment:Coal [J]  ! RunPeriod
64,ExteriorEquipment:FuelOil#1 [J]  ! RunPeriod
75,ExteriorEquipment:FuelOil#2 [J]  ! RunPeriod
86,ExteriorEquipment:Propane [J]  ! RunPeriod
97,ExteriorEquipment:Water [M3]  ! RunPeriod
End of Data Dictionary
     1,CHANUTE AFB ILLINOIS SUMMER,  40.30, -88.13,  -6.00, 229.51
     5,  1
26,2.4441696E+10,0.0000000E+00, 7,21, 1,01:10,1.0543500E+08, 7,21,18,51:60
42,8.6400000E+07,600000.0, 7,21, 1,01:10,600000.0, 7,21, 1,01:10
53,8.6400000E+07,600000.0, 7,21, 1,01:10,600000.0, 7,21, 1,01:10
64,8.6400000E+07,600000.0, 7,21, 1,01:10,600000.0, 7,21, 1,01:10
75,8.6400000E+07,600000.0, 7,21, 1,01:10,600000.0, 7,21, 1,01:10
86,8.6400000E+07,600000.0, 7,21, 1,01:10,600000.0, 7,21, 1,01:10
97,9.5040000E+08,600000.0, 7,21, 1,01:10,6000000., 7,21, 1,01:10
301,2.4355291E+10,0.0000000E+00, 7,21, 1,01:10,1.0543500E+08, 7,21,18,51:60
306,7.0114299E+09,0.0000000E+00, 7,21, 1,01:10,1.0543500E+08, 7,21,18,51:60
311,7.7724027E+09,0.0000000E+00, 7,21, 1,01:10,5.9570756E+07, 7,21,24,51:60
316,9.5714509E+09,0.0000000E+00, 7,21, 1,01:10,1.0543500E+08, 7,21,18,51:60
```

```
26,2.4441696E+10,0.0000000E+00, 7,21, 1,01:10,1.0543500E+08, 7,21,18,51:60
```

The previous line shows the output for #26, the Electricity:Facility meter for the entire run period (which is just a summer design day). The second value 2.4441696E+10 is the recorded usage for that run period. The third value 0.0000000E+00 is the minimum value recorded for the reporting period along with the timestamp information in the following 4 fields "7,21, 1,01:10," – July 21st, hour 1, and 1:10 (minute period in the hour when the minimum occurred). Then follows the maximum value for this meter (1.0543500E+08) along with its timestamp information " 7,21,18,51:60" – July 21st, 18th hour, minutes 51:60.

Facility meters contain all the energy of a fuel type. Building meters contain the sum of each zone's energy. System meters contain the energy from the system components. Plant meters contain the energy from the plant equipments.

Thus, the following relationships should be observed:

$$Facility = \sum \left(Building + System + Plant + Exterior \right)$$

$$Building = \sum_{1}^{NumberOfZones} Zone$$

Meter names are of two forms:

<FuelType>:<name>

or

<EndUseType>:<FuelType>:<name>.

The user requests reporting by either specifying the full name above (without Units) or a "wildcard" represntation of that name (to obtain all meters of a type). For example, entering "Electricity:*" for the meter name will report on all the Electricity: meters (Electricity:Facility, Electricity:Building, Electricity:Zone:Resistive Zone, Electricity:Zone:East Zone, Electricity:North Zone in the example above). Both the fuel types and enduse types are set within the program by the developers. Current Fuel types are shown in the table below:

Table 11. Table of Metered Fuel Types

Energy/Fuel Type
Electricity
Gas
Coal
FuelOil#1
FuelOil#2
Propane
Water

The end use types are shown in the following table:

Table 12. End Use Category Types

End Use Types
GeneralLights
TaskLights
ExteriorLights
Heating
Cooling
DHW
Cogeneration
ExteriorEquipment
ZoneSource
PurchasedHotWater
PurchasedChilledWater

Index

B

bugreprt.txt, 8
Building Simulation, 19
- Checklist, 19
- Model Construction, 22
  - Compile Information
    - Building, 27
    - Subsurface, 27
    - Surface, 27
    - Zone, 27
  - Compile Space Gain Data, 29
  - Construction Elements, 26
  - Equivalent Surfaces, 25
  - Surface Types, 22
- Zone, 19

D

Data Files, 42

E

Energy Meters, 43
Energy+.idd, 8
EnergyPlus.exe, 8
EP-Launch, 9
EP-Macro.exe, 8

I

IDFEditor, 8
Index, 46
Input Files, 14
- Energy+.idd, 14
- Energy+.ini, 14
- in.epw, 14
- in.idf, 14

Input Files, Create, 31
- BLASTTranslator, 36
- DOE2Translator, 37
- EnergyPlus Units, 33
- IDD Conventions, 31
- IDFEditor, 34
  - Caveats, 36
  - Change Values, 36
  - Class List, Objects, 35
    - Working with, 36
  - Create/Select IDF, 35
  - Edit Menu, 36
  - File Menu, 36
  - Start, 35
  - View Menu, 36
- IFCtoIDF Utility, 38
- Text Editor, 38, 39

M

Meter
- End-Use-Categories, 45
- Fuel-Types, 45
- Types, 44

O

Output Files, 14
- audit.out, 15
- eplusout.dbg, 16
- eplusout.dxf, 16
- eplusout.eio, 16
- eplusout.err, 14
- eplusout.eso, 15
- eplusout.rdd, 16
- eplusout.sln, 16

P

Post Processing, 41
Postprocessor Files, 16
- *.rvi, 18
- eplusout.csv, 18
- eplusout.eso, 17
- eplusout.inp, 17

R

readme file(s),, 8
RunEPlus, 9, 12
RunEPlus.bat, 8

W

Weather Data, 40