

# Information Technology Basics for Energy Managers— How a Web-based Energy Information System Works

*Barney Capehart, University of Florida*

*Paul Allen, Reedy Creek Energy Services*

*Klaus Pawlik, Accenture*

*David Green, Green Management Services Inc.*

## ABSTRACT

Advances in new equipment, new processes, and new technology are the driving forces in improvements in energy management, energy efficiency, and energy cost control. Of all recent developments affecting energy management, the most powerful new technology to come into use in the last several years has been information technology—or IT. The combination of cheap, high-performance microcomputers together with the emergence of high-capacity communication lines, networks, and the internet has produced explosive growth in IT and its application throughout our economy. Energy information and control systems have been no exception. IT and internet-based systems are the wave of the future. Almost every piece of equipment and almost every activity will be connected and integrated into the overall facility operation in the next several years.

The internet, with the worldwide web—or web—has become quickly and easily accessible to all facility employees. It has allowed the development of many new opportunities for energy and facility managers to quickly and effectively control and manage their operations. The capability and use of IT and the internet in the form of web-based energy information and control systems continues to grow at a very rapid rate. New equipment and new suppliers appear almost daily, and existing suppliers of older equipment are beginning to offer new web-based

systems. Facility managers, maintenance managers, and energy managers are all having to deal with this rapid deployment of web-based equipment and systems, and need to be prepared for current and future applications of internet-based technologies in their facilities. In some cases, facilities are developing their own information and control systems or at least subsystems, and are trying to understand how to connect and interface new IT equipment to their older energy management or facility management systems.

The purpose of this article is to help prepare energy managers to understand some of the basic concepts and principles of IT. We hope that they can successfully apply IT to their facility, and have the knowledge to supervise the IT work of a consultant or a vendor. Knowing what is going on and what is involved is important information for energy managers if they are going to successfully purchase, install, and operate these complex, web-based energy information and control systems. Energy management is a very comprehensive area, and this article is the first in a series whose purpose is to address the most significant concepts and principles that the typical energy or facility manager might need. The emphasis of this series is on computer networking, use of facility operation databases, and sharing data using the web and the TCP/IP communications protocol. This first article will introduce basic principles, structures, and definitions needed for most facility IT applications. Future articles will cover general application principles and discuss specific software and hardware requirements for typical energy information and control systems.

## ENERGY INFORMATION SYSTEM (EIS)

The philosophy, “if you can measure it, you can manage it,” is critical to a sustainable energy management program. Continuous feedback on utility performance is the backbone of an energy information system. A basic definition of an energy information system is:

### **Energy information system (EIS)**

Equipment and computer programs that let users measure, monitor, and quantify energy usage of their facilities and help identify energy conservation opportunities.

There are two main parts to an EIS: (1) data collection and (2) web publishing. Figure 1 shows these two processes in a flow chart format.

Everyone has witnessed the growth and development of the internet—the largest computer communications network in the world. The internet and the worldwide web (web), using the TCP/IP communications protocol, has made it much easier to access and distribute data. Using a web browser, one can access data around the world with a click of a mouse. An EIS should take full advantage of these new tools.

### **Intranet**

An intranet is a small, self-contained, private version of the internet, using internet software and internet communications standards. Companies are increasingly using intranets to give their employees and other approved people easy access to facility and corporate data.

### **TCP/IP**

Transmission control protocol/internet protocol (TCP/IP) is a family of industry standard communications protocols that allow different networks to communicate. It is the most complete and accepted enterprise networking protocol available today, and it is the communications protocol of the internet. An important feature of TCP/IP is that it allows dissimilar computer hardware and operating systems to communicate directly.

### **Web Browser**

A web browser is a program used to display data transferred over an intranet or the internet. It lets users select, retrieve and interact with resources on the web. The most commonly used web browsers are Internet Explorer and Netscape.

## **METERING EQUIPMENT**

The first task in establishing an EIS is to determine the sources of the energy data. Utility meters monitored by an energy management system or a dedicated power-monitoring system are a good source. These systems provide a good means to collect energy data. Metering equipment collects the raw utility data for electric, chilled water, hot water, natural gas, and compressed air. Information from these meters is collected by either direct analog connections, pre-processed pulse outputs, or by digital, network-based protocols.

### **Direct Analog Connection**

The utility meter produces a 4-20mA or 0-10V signal for the instantaneous usage continuously monitored by an energy management or process control system. The control system calculates the utility usage integrating the instantaneous values over time.

### **Pulse-Output**

The utility meter outputs a pulse for a pre-defined amount of energy usage. A local data storage device accumulates the pulses.

### **Digital, Network-based Protocol**

The utility meter calculates and totals the energy used within the meter. A dedicated local area network (LAN) links the energy meters together. A local data storage device polls these meters at predefined intervals to store the energy data until retrieved by the server.

## **DATA COLLECTION PROGRAM**

Data gathered from all of the local data storage devices at a pre-defined interval (usually on a daily basis) are stored on a server in a relational database.

### **Relational Database**

A relational database is a collection of tables, rows, and columns used to store data, and is organized and accessed according to relationships between data. Examples of relational databases are FoxPro, SQL, and Oracle.

There are a variety of methods used to retrieve this data:

### **Modem Connection**

A modem connection uploads the data from the local data storage device to the energy data server. Typically, the upload takes place on a daily basis, but could be transferred more frequently, if needed.

### **LAN or WAN Network Connection**

A local area network (LAN) or a wide area network (WAN) connection, established between computers, transfers energy data files to the energy data server.

### **FTP Network Connection**

File transfer protocol (FTP) is an internet protocol used for transferring files from one computer to another. It is used to move the energy data files from the local data storage devices to the energy data server.

Once the energy data has been transferred to the energy data server, an update program reads all of the various data files and reformats them into a format that is used by the web publishing program.

An essential feature of this process is to design the programs and tables so they will update quickly. Typically, utility meters can generate an enormous amount of data, so it is important to keep the table records manageable. Breaking the data up into yearly, monthly, weekly, and daily files keeps the data manageable.

## **WEB PUBLISHING PROGRAM**

To publish the energy data on an intranet or the internet, client/server programming is used. The energy data is stored on a central computer, the server, and waits passively until a user makes a request for information using a web browser, the client. A web publishing program retrieves the information from a relational database, sends it to the web server, which then sends it to the client that requested the information.

### **Web Server**

A web server is a program that runs on a network server (computer) to respond to HTTP requests. The most commonly used web servers are Internet Information Server and Apache.

### **HTTP**

Hypertext transfer protocol (HTTP) is an application layer protocol used to transfer data across an intranet or the internet. It is the standard protocol for moving data across the internet.

### **HTTP Request**

An HTTP request is data sent from a web browser to a web server.

### **HTTP Response**

An HTTP response is web browser content sent from a web server to a web browser in response to a specific HTTP request.

A CGI interface program coordinates the activity between the web-server and the web publishing program and allows for simultaneous multi-user access.

### Common Gateway Interface (CGI)

CGI is a method used to run conventional programs through a web browser.

The client/server process for an EIS uses the steps below (See Figure 1):



Figure 1. Energy Information System Functional Layout

1. A user requests energy information by using their web browser (client) to send an HTTP request to the web server.
2. The web server on the network server activates the CGI interface program. The CGI program then starts up the web publishing program.
3. The web publishing program retrieves the information from the relational database, formats the data in HTML and returns it to the CGI interface program.
4. The CGI interface program sends the data as HTML browser content to the web server, which sends the content to the web browser requesting the information.

This entire process takes only seconds, depending on the connection speed of the client's computer to the web.

## **HTML**

Hypertext markup language (HTML) is the format of the file containing web browser content. Developers use HTML to create web pages, and build graphical documents that contain images, formatted text, and links to other documents on the web.

## **Browser Content**

Browser content is data—text, numbers, images, and links—displayed in a web browser. Content is the new name for information carried over a network. It may be words, numbers, pictures, sound, or any form of what we used to call information.

## **PROGRAMMING CHOICES FOR EIS WEB PUBLISHING**

The following sections provide additional detail on programming choices used in the development of the web publishing programs. These are only a few of the many programs available. Both server-side and client-side programming environments are applicable.

### **Using Server-Side CGI Programs to Create Content**

Server-side programs are programs that run on the network server. CGI programming is the most fundamental way to access relational database information over the internet or intranet and display dynamic

content. This is important to any EIS, since the amount of data involved will undoubtedly require processing with a relational database of some kind.

### **Dynamic Content**

CGI programs create dynamic content as a web page that is displayed in a web browser at the same time it is created.

### **Static Content**

Developers create static content as a web page that is stored in a file and displayed later. The web browser retrieves the data and displays the static content.

Although CGI is a necessary piece of the puzzle in creating dynamic content, it may just be a stepping-stone to some other program. Small CGI programs oftentimes just launch secondary programs that run on the web server and return content to the browser. A secondary program could be nearly any custom program that has the ability to send data back to the web server. *FoxWeb* is a small CGI program that connects to a Visual FoxPro database application. The application then calls any number of custom designed queries and procedures to return results to the browser as HTML. More information about FoxWeb is available at <http://www.foxweb.com>.

*PERL* (practical extraction and report language) is an application used for CGI programming. One reason is that it is free. Another is that it is portable across operating systems. In other words, applications written in PERL will work on any operating system. It also has the ability to connect to many types of databases. A good source for information on PERL is at <http://www.perl.com>.

*ColdFusion* is a server-side application that uses a ColdFusion server. The server executes templates containing a mixture of HTML and ColdFusion instructions, and then returns the results to the browser as pure HTML.

*Active Server Pages (ASP)* are also a popular choice recently. The ASP program on the Windows web server will automatically interpret web pages ending with the extension “.asp.” The web pages are a mixture of ASP instructions, Visual Basic code and HTML. ASP is not portable across operating systems. It only works with Microsoft web servers.

*Java Servlets* are Java programs that run on a web server and



build web pages. *Java* is the latest of a long line of “higher level” programming languages such as FORTRAN, Pascal and C++. It is also portable across operating systems. Java Servlets written for one web server on one operating system will run on virtually any web server and operating system.

*Java Server Pages (JSP)* are similar to *Active Server Pages* except that the pages consist of Java code in place of Visual Basic code. This makes the code portable across operating systems. The web pages typically end with the extension “.jsp.” This tells the web server to interpret the embedded *Java* code like any other *Java Servlet*. Information about *Java Servlets* and *Java Server Pages* is available at this Johns Hopkins University web site: <http://www.apl.jhu.edu/~hall/java/>.

Any organization wishing to develop an EIS should carefully consider which server-side applications to use. The decision should be a practical one rather than a popular one. All of the criteria below should be a part of the evaluation process:

- What operating system is predominantly available to the facility?
- What programming languages are the support personnel willing to work with?
- What applications are compatible with the existing database?
- How much of the budget is available to spend?

### **Using Client-Side Applications to Enhance Content**

Client-side applications can create a deeper level of interactivity within web pages. Scripting languages such as *JavaScript* and *VBScript* are less complex versions of other languages like Java and Visual Basic, respectively. They reside within the HTML of a web page and provide a great deal of functionality that HTML itself cannot. Scripts such as these can validate input, control the cursor programmatically and much more.

*Dynamic HTML (DHTML)* is the result of scripting languages taking advantage of the extensions common to the latest browsers to make the pages change after they are loaded. A good example of this is a link that changes color when the user places the mouse cursor over it. Much more dramatic effects are possible using DHTML. However, the two most popular browsers, Internet Explorer and Netscape, interpret DHTML differently. Good information about DHTML is available at <http://www.dynamicdrive.com/>.

*Cascading Style Sheets (CSS)* are special HTML features that allow much more flexibility to format elements of a web page. The ability of CSS to describe the style of an element only once rather than every time you display the element provides a separation of content and presentation. This makes your web pages less complex, much smaller and therefore faster to load. Beware that CSS is only fully supported in the latest versions of browsers (4.0 and above).

*Java Applets* are small Java programs that are stored on a web server and are called for from the HTML in a web page. Statements in the HTML pass parameters to the applet affecting its functionality. Unlike *Java Servlets*, the browser downloads the applet and runs it using the browser operating system rather than the operating system of the web server. Free Java applets are widely available on the internet. KavaChart applets are very useful for charting trends in data. KavaChart applets are available on-line at <http://www.ve.com>.

*Extensible Markup Language (XML)* XML is a meta-language and has a number of uses. A meta-language is a language used to explain another language. XML organizes data into a predefined format for the main purpose of sharing between or within computer systems. Furthermore, its uses include data organization and transfer, data presentation, data caching, and probably some that we have not invented yet. More information about XML is available at <http://xml.com/>.

Developers can use any or all of these to enhance the content of an EIS. Three important points to remember about using client-side applications to enhance your browser content are:

1. Many client-side applications require later versions of browsers to work correctly. Be sure all of your users are using the required browser versions.
2. Many client-side applications are available free. Search the internet before spending resources developing your own custom client-side applications.
3. Client-side applications will make your web pages more complex. This adds to development and maintenance costs. Be sure to weigh the benefits of these enhancements against their costs.

### **Support for EIS Web Publishing**

Support for an EIS is quite involved. It includes help pages on the application itself to guide the user. There must be an e-mail address and telephone number of someone for users to contact. Adequate system

support personnel must be available to help with system or network problems. Developers should track and document updates to the application. Undoubtedly, at some point, the tools you use to develop your EIS will require support as well.

Help pages should be simple and to the point. Avoid long discussions about how the EIS is developed and why. Save that for the "About This System" section or something similar. Each menu item or function should have its own help page. That information should be accessible from the main menu or at anytime within the function itself. Contact information about where to get further help should be available on every page.

E-mail addresses and telephone numbers published for help should be those of the local data expert. Many problems arise due to missing data or misunderstanding of the data.

These issues should be resolved or discounted before sending the problem on to the programming or system support personnel.

IT system support may not be as responsive as we would like. These personnel have a tremendous job to do. It is important to maintain a good relationship with them since their expertise is critical to the success of the EIS. However, at times when they are not able to respond it helps to have as much control over the EIS system as possible. Perhaps locating the server or the data collection machine in a location that is more accessible can help. Many times simply re-running a data collection routine or making a minor change to the code is all that is needed to solve a problem.

The programming support personnel should track and document all changes to the code. A good version tracking system is necessary to preserve versions that may be useful later. Keeping a maintenance log will help to recall changes made to the EIS system as well as contact information for sources of outside help.

Outside support from vendors comes in various forms. There is "free," "free only by e-mail," "paid by hour," "paid by the call," and many other variations. Due to the nature of the EIS data, these applications require a constant level of support. Not necessarily a great deal of support, but they seem to be in a constant state of evolution. There may be a new report for someone, new data available, anomalies in the data or new versions of tools to install. The value of outside support that is available for the tools used may be a deciding factor in choosing components of the EIS.

## Choosing EIS Web Publishing Tools

We might define “tools,” in this case, as utility applications that require more configuration effort than programming effort. There are some exceptions of course. In any case, tools fall into three categories:

- 1) open-source or free
- 2) purchased
- 3) developed

Tools are needed to perform such functions as batch e-mailing, charting, scheduling application run times, and database to web connectivity. The relational database and web server are also tools.

Batch e-mail applications can be of any category. There are good free ones and purchased ones. Some e-mail servers have batch processing capability and some do not. Purchased batch e-mail applications are relatively inexpensive. They have some variations in features such as whether or not they will send HTML or attachments. They are also easy to develop.

Charting tools are really too complex to warrant developing. There are some very good free and open-source charting tools that have all the features of purchased ones. If not, one could probably purchase extensions to the standard features of a free charting tool.

The EIS needs scheduling programs to launch data collection applications at predefined times. Some operating systems have scheduling programs built in. They may be difficult to configure though. A purchased version is probably the best choice since their cost is quite low.

There are some applications that do much of the database to web connectivity for you. These require either purchasing or, in the case of the free ones, a great deal of programming. The purchased ones are a good choice since much of the error reporting is part of the application. The purchased versions can be very expensive or relatively inexpensive. Each database has its own connectivity options so much of this decision rests on which relational database is used.

The database used is likely a purchased one or may actually be open-source. There are open-source databases like MySQL competing with the best of the others. Commercial relational database systems range in cost from very inexpensive like MS Access to very expensive like Oracle.

Open-source web servers like Apache are available for some oper-

ating systems and are very widely used and reliable. Others are free with the operating system like MS IIS. Some are also commercially available for a few hundred dollars. The web server choice depends mostly on the operating system of the server itself.

The EIS tools will likely be a mix of open-source, purchased and developed applications. It is important to consider budget constraints, operating systems and support when deciding which and what type of tools to use. Always plan ahead of time for the compatibility of all the tools the EIS will use.

## ENERGY INFORMATION SYSTEM EXAMPLE

In 1997, the utility reporting system (URS) was developed by Reedy Creek Improvement District (RCID) to provide a means to “publish” utility metering information and track the results of energy saving efforts at Walt Disney World using the Disney intranet. The URS provides continuous feedback on utility performance and pinpoints energy waste for further investigation.

This section will “get under the hood” and describe the specific programming tools used to create the URS. The goal here is to relate some of the basic concepts presented earlier in the article to a specific example of an energy information system.

### Relational Database Table Structure

The URS uses a relational database program called Microsoft Visual FoxPro. The first step in the URS development was to organize the data table structures. Below are the basic tables:

*Account definition table:* Includes meter account level definitions and grouping variables used by the program.

*Monthly data tables:* Includes the monthly billing data from RCID utility billing system.

*Daily data tables:* Includes data recorded on an hourly basis from power monitoring systems and energy management systems.

*E-mail data table:* Includes names, e-mail addresses, and the utility reports sent.

### Data Collection Programs

The next step in the development of the URS was to develop pro-

grams to pull the utility data into the data tables. Visual FoxPro reads the various data sources and organizes the data into common data tables. Shown below are the various data collection tasks in the URS:

*Monthly Utility Data:* An FTP transfer from the RCID billing system downloads this data in an ASCII comma delimited file on a monthly basis. The database administrator creates a new record in the account definition table each time a new billing account is available.

*Power Monitoring Hourly Data:* RCID's supervisory control and data acquisition (SCADA) power monitoring system records max, min, and average hourly data for meters and outputs this data to an ASCII file each day. A Visual FoxPro program reads this data and reformats it into hourly data tables.

*Energy Management System (EMS) Hourly Data:* The EMS produces files that include trends of analog/digital points and consumable data from utility meters connected to the EMS. Data collection programs copy these files from the EMS servers to the URS Server after the EMS creates them each night. A Visual FoxPro program reads the data from these reports and reformats it into the hourly data tables.

*Veris Industries Power Monitoring Data:* Hourly data is recorded in local data collection devices called Eservers. On a nightly basis, a program supplied by Veris Industries automatically collects this data from each Eserver into an ASCII comma delimited file. A Visual FoxPro program pulls the data from these reports and into the hourly data tables.

Once the data collection programs collect all of the data from the various data sources, a Visual FoxPro program updates all of the URS data tables. Once completed, the program copies the standard data file tables to the URS server data directory. The data is now ready for viewing on the web.

## **E-mail Utility Reports**

To make the URS easy to use, it sends HTML-based reports via e-mail on a daily basis (to report on hourly data collected) and a monthly basis (to report on monthly billing data). E-mail increases the likelihood that the user views the utility data. Instead of waiting for the user to visit the URS web site and figure out how to generate the same report, the URS delivers the report via e-mail.

By creating an HTML-based report, links to graphs can be embedded into the report to make it more visual (see Figure 2). Users view the reports using their e-mail program (Microsoft Outlook) and are able to

produce graphs by simply clicking on links in the e-mail. Sending mail on utility usage helps to increase employee participation in reducing their facility's energy consumption.

### Web Publishing Program

The program language used to generate the web pages for the URS is Visual FoxPro. FoxWeb is the CGI program that interfaces Visual FoxPro with the web server. Different reports allow viewing and graphing both the monthly utilities billing information and the hourly utility data. Kavacharts Java applets, called from the Visual FoxPro programs, generate the graphs used in the URS.

The challenge of producing an effective EIS is to create reports that are both informative and easy to use. The URS makes extensive use of

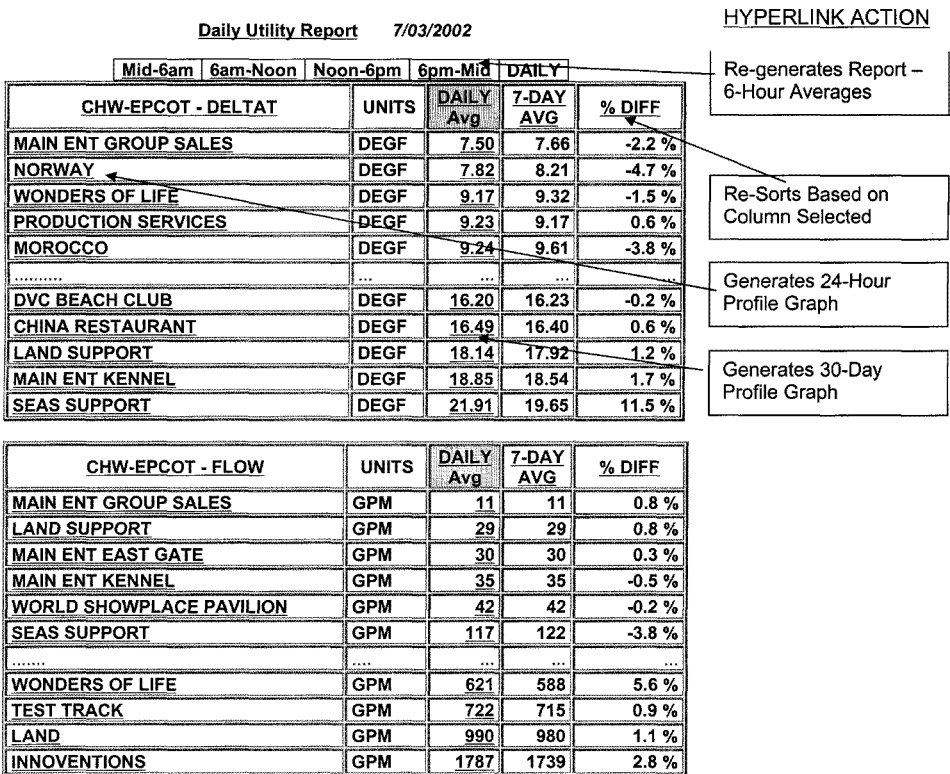


Figure 2. Utility Reporting System HTML-based Report Example Daily Chilled Water Report

embedded links to sub-reports and graphs. Most reports in the URS show the data in tabular format with embedded links on numbers and data labels that the user can click on to either re-sort the data, produce detailed sub-reports or graphs showing the data various ways. This makes the use of the URS intuitively easy for the user.

## CONCLUSION

Today's energy manager needs to be knowledgeable of the basic principles and concepts of IT, since this is a fast-growing area of new systems and services. This first article described the basics of how an energy information system is set up and how it works. Future articles on related energy management topics will help de-mystify the IT terms associated with them. A future article will also address the general principles and concepts associated with an energy control system, since the foundation has been laid with this discussion of an energy information system.

## ADDITIONAL INFORMATION

More information about energy information systems as well as links to many of the tools discussed in this article can be found at <http://www.utilityreporting.com/>.

### **Bibliography**

- Allen, Paul J and David C. Green, "Managing Energy Data Using an Intranet—Walt Disney World's Approach," Proceedings of the 2001 World Energy Engineering Congress Conference & Expo, Atlanta, Ga., October, 2001.
- Bos, Bert, "Cascading Style Sheets" [article on-line] (2002, accessed 8 July 2002); available from <http://www.w3.org/Style/CSS/>; Internet.
- Hall, Marty, "Java Programming Resources" [article on-line] (1999, accessed 8 July 2002); available from <http://www.apl.jhu.edu/~hall/java/>; Internet.
- INT Media Group, Incorporated, "TCP/IP" [article on-line] (2002, accessed 6 June 2002); available from [http://webopedia.intemet.com/TERM/T/TCP\\_IP.html](http://webopedia.intemet.com/TERM/T/TCP_IP.html); Internet.
- Ireland, Blair, Introduction to Perl—Become a Guru [article on-line] (accessed 6 June 2002); available from <http://www.thescripts.com/serversidescripting/perl/tutorials/introductiontoperl—becomeaguru/page0.html>; Internet.



- Kington, Max, "ColdFusion—An Introduction" [article on-line] (accessed 6 June 2002); available from <http://www.thescrpts.com/serversidescripting/coldfusion/tutorials/coldfusion-anintroduction/>; Internet.
- Marshall, James, "HTTP Made Really Easy" [article on-line] (August 15, 1997, accessed 8 July 2002); available from <http://jmarshall.com/easy/http/>; Internet.
- Murdock, Robert, "ASP Basics—What is ASP?" [article on-line] (accessed 6 June 2002); available from <http://thescrpts.com/serversidescripting/asp/tutorials/aspbasics/page0.html>; Internet.
- NCSA Software Development Group, "CGI: Common Gateway Interface" [article on-line] (1999, accessed 6 June 2002); available from <http://www.w3.org/CGI/>; Internet.
- Richmond, Alan, "Dynamic HTML" [article on-line] (2002, accessed 6 June 2002); available from <http://wdvl.internet.com/Authoring/DHTML/>; Internet.

### Software References

- "Apache HTTP Server Project," Apache Software Foundation, <http://www.apache.org/>
- "ColdFusion MX," Macromedia, Inc., <http://www.macromedia.com/software/coldfusion/>
- "Eserver Energy Information Server," Veris Industries, Inc., Portland, Oregon, <http://www.veris.com>
- "FoxWeb," Eon Technologies, Alameda, California, <http://www.foxweb.com>
- "Java," Sun Microsystems, Inc., <http://java.sun.com/>
- "Kavacharts," Java Applet for graphing data, <http://www.ve.com>
- "MySQL," MySQL AB, <http://www.mysql.com/>
- "Perl," Perl Mongers —The Perl Advocacy People, <http://www.perl.org/>
- "Visual FoxPro," Microsoft Corporation, <http://msdn.microsoft.com/vfoxpro/>
- Richmond, Alan, "Dynamic HTML" [article on-line] (2002, accessed 6 June 2002); available from <http://wdvl.internet.com/Authoring/DHTML/>; Internet.

---

### ABOUT THE AUTHORS

**Barney Capehart** is a professor emeritus at the University of Florida College of Engineering, and is a member of the AEE Hall of Fame. He regularly teaches five-day Energy Management Training Seminars for AEE. Barney is the lead author of *Guide to Energy Management* from Fairmont Press, and is a fellow of IEEE, IIE and AAAS. Barney was the director of the University of Florida Industrial Assessment Center for ten years, and was the recipient of the 1988 Palladium Medal from the American Association of Engineering Societies. [Capehart@ise.ufl.edu](mailto:Capehart@ise.ufl.edu)

**Paul Allen** is the chief energy management engineer at Reedy Creek Energy Services (a division of the Walt Disney World Co.) and is responsible for the development and implementation of energy conservation projects throughout the Walt Disney World Resort. Paul is a graduate of the University of Miami (BS degrees in physics and civil engineering) and the University of Florida (MS degrees in civil engineering and industrial engineering). Paul is also a registered Professional Engineer in the State of Florida. The Association of Energy Engineers (AEE) selected Paul as the 2001 Energy Manager of the Year. ([paul.allen@disney.com](mailto:paul.allen@disney.com))

**David Green** has combined experience in intranet/internet technology and database queries and has developed programming for energy information systems. David has been the president of his own consulting company, Green Management Services, Inc., since 1994. He has a Bachelor of Science degree in chemistry and a Master of Arts degree in computer science. David is also a lieutenant colonel in the Illinois Army National Guard and has 18 years of military service. David has successfully completed major projects for The ABB Group, Cummins Engine Company, ECI Telematics, the M.A.R.C of the Professionals and The Illinois Army National Guard. ([dcgreen@dcgreen.com](mailto:dcgreen@dcgreen.com))

**Klaus Pawlik** is a consultant with Accenture working in the utilities industry. Klaus is the author of the *Solution Manual for Guide to Energy Management, Third Edition*. Klaus holds a master's in business administration and a Bachelor of Science in industrial and systems engineering graduating with highest honors from the University of Florida. While at the University of Florida, Klaus worked in the Industrial Assessment Center leading teams of undergraduate and graduate students performing energy and waste minimization, and productivity improvement assessments for manufacturing facilities. Additionally, for two years, he assisted Dr. Barney Capehart with teaching industrial energy management. Before attending the University of Florida, Klaus served six years in the United States Navy, where he worked as an electrical operator on nuclear power plants. For two of those years, he served as an instructor training personnel on the electrical operations for nuclear power plants. ([klaus.e.pawlik@accenture.com](mailto:klaus.e.pawlik@accenture.com))