Utility Data Web Page Design—An Introduction

David C. Green
Green Management Services, Inc.

Paul J. Allen
Walt Disney World

ABSTRACT

In order to make a conscious effort to conserve energy at large-scale facilities it is necessary to track utility use on a daily basis. An added benefit of utility tracking is that discovering and troubleshooting equipment malfunctions becomes much easier. Utility data have unique and complex characteristics. Collecting, analyzing, and publishing utility data to a large number of people puts more eyes on the task of managing utilities. Different people use the data in different ways. Internet technologies allow us to publish utility data, in a meaningful way, to all of those people simultaneously.

INTRODUCTION

Utilities seem to be the substance of our existence. In almost every minute of our busy lives we consume some sort of utility. Electricity, water, natural gas, fuels, and others all play an important part in our lives. To track the use of these commodities minute by minute is a task vital to the goal that we conserve and use these resources as efficiently as possible. The amount of utility data collected everyday is vast. The data itself are somewhat complex in nature due to the multitude of utility types and means of measurement.

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. At the center of today's energy information systems is the web technology that publishes data to a wide audience on a daily basis. Energy managers, technicians, department managers, and directors all become involved in utility monitoring by using these systems. (1)

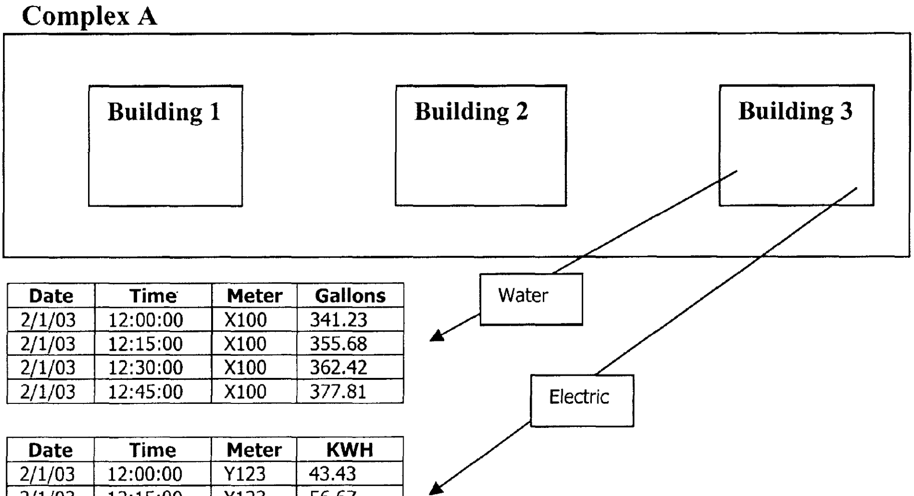
The capability and use of information technologies and the internet in the form of web-based energy information and control systems continues to grow at a very rapid rate. To publish the utility data on an intranet or the internet, client/server programming is used. The energy data is stored on a central computer, the server, and waits passively until a user, the client, makes a request for information using a web browser. 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. (1)

The purpose of this article is to introduce how to publish utility data using a web page. We explain the characteristics of utility data, who needs to see the data and why. Then we explain the web page design requirements and introduce some of the internet technologies used to present utility data so that it is most meaningful.

CHARACTERISTICS OF UTILITY DATA

Utility data have unique characteristics that make web applications especially suited to reporting it. First, utility data is usually collected on an hourly basis, if not more often. This creates a huge amount of data. Only a powerful database application can store and manipulate that much data. Systems collect data from many different locations and report it on different levels. The data are usually date specific and require trending over time to be useful. The data come from many types of utilities and are collected in a variety of units of measure.

Systems collect utility data quite often from many different locations. It may need to be broken down into groups for reporting purposes. Looking at the utility usage for different "levels" is a reasonable expectation. The electricity consumption of a building means just as much as for a whole complex of buildings. It might also be necessary to combine data from several meters and report it as one data point or fractionalize data from one meter and report it as more than one data point. Systems record all data with a time stamp. Below is a diagram of a typical complex of buildings and the some of their data.



Date	Time	Meter	Gallons
2/1/03	12:00:00	X100	341.23
2/1/03	12:15:00	X100	355.68
2/1/03	12:30:00	X100	362.42
2/1/03	12:45:00	X100	377.81

Date	Time	Meter	KWH
2/1/03	12:00:00	Y123	43.43
2/1/03	12:15:00	Y123	56.67
2/1/03	12:30:00	Y123	67.87
2/1/03	12:45:00	Y123	79.98

Figure 1. Data collection

The data are only meaningful when presented in the context of time. Hourly data for specific days and daily data spread over the period of a month provide the most meaningful reports and graphs. Comparisons to other data, such as outside temperature, are also helpful. Sometimes averaging data or summing for each utility is required. Systems can average or sum data together over periods of time or by combining meters at various locations.

Below is a graph of average hourly electric consumption for all of Building 1 over a 24-hour period, along with average outside temperature.

Below is a graph of total daily electric consumption for one meter in Building 1 over a 30-day period.

A typical facility can produce data for every electric, water, and gas meter, as well as others. Temperature sensors collect temperature data. Different utilities produce data in different units of measure. Tracking the units at the lowest level is important so that applications sum and average data over time correctly. It is important to keep in mind that the units of measure may change as metering requirements or reporting requirements change. Table 1 shows some examples of utility data units of measure.

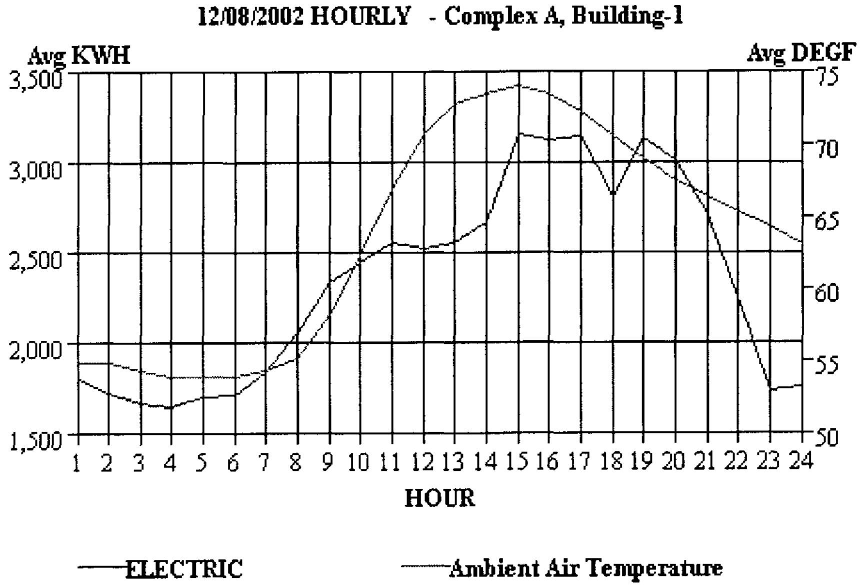


Figure 2. Average hourly electric consumption vs. average outside temperature

Table 1. Utility data units of measure

UTILITY	*UNITS OF MEASURE*
Hot Water	MMBtu, kBtu
Cold Water	gal, kgal
Natural Gas	kscf
Electric	kWh, MWh
Compressed Air	kscf, scfm
Flow	gpm
Pressure	psi
Fuel oil	gal
Chilled Water	tonhrs
Temperature	Degf degc
Efficiency	kW/ton

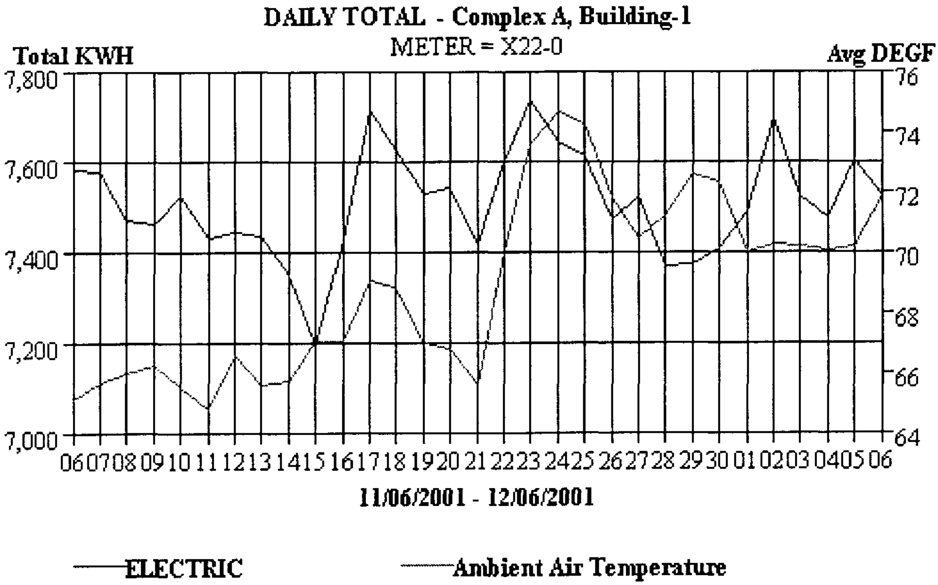


Figure 3. Total daily electric consumption vs. average outside temperature

USERS OF UTILITY DATA

Users of utility data are very diverse. A director of a facility might find utility data just as useful as a technician would. Energy managers look at this data perhaps more than any others do. An energy manager might look for spikes in the data signifying a mechanical failure. Department managers will look for conservation opportunities in hourly data and track their performance month to month. Directors may be interested in some critical data affected by temperature extremes or economic circumstances. Technicians typically respond to evaluations of data results but may find it helpful to "zoom in" on data to get an idea about what a particular piece of equipment is doing.

Energy managers must communicate with maintenance and operations personnel on a daily basis. It is helpful in communicating to know the performance data of the equipment and systems involved. Energy managers use the data to evaluate the performance of systems such as HVAC. They compile, maintain, and distribute reports from the data.

These reports help identify conservation opportunities and the effectiveness of conservation measures already taken. They also may use the data to identify equipment malfunctions.

Department managers set procedures in place to conserve energy and then use the utility data to monitor the progress of those procedures. They can identify places and times to improve conservation efforts. Monitoring utility data also provides them an opportunity to watch for abnormalities in equipment performance.

Directors use utility data to track critical areas of concern that may affect budget requirements. They try to use the data to predict the future impact of certain conditions. They plan according to trends in data. Directors might use the data to report on reasonable expectations of costs for future development.

Technicians typically respond to action requests brought on by evaluating the utility data. They then use the data to monitor detailed system functions and make adjustments accordingly. They may use the data to set procedures for operating and maintaining the equipment properly.

WEB PAGE DESIGN REQUIREMENTS

Web pages that display utility data must be informative, intuitive, reliable, and robust. The display should have as much data as can fit on the page and still be readable and printable. Pages that provide as many ways as possible to change the presentation of the data quickly and easily are robust. A good intuitive design provides results with as few selections as possible. Developers should create pages in such a way as to be compatible with many types of browsers.

The vast amount of data involved with utility reports precludes the need to display as much as possible at a time. The source of the data, as well as its utility type and units, must be displayed. Other values to show costs, averages, efficiencies, etc. are all more meaningful when set alongside the data values themselves in one report. Figure 4 shows how this report might look.

Sorting and filtering from links in the display gives the user a quick and easy way to change the presentation with as few selections as possible. Option links provide the ability to change values instantly. These methods provide an intuitive utility data web page design.

Utility Report 12/08/2002

COMPLEX A

BUILDING-1 - ELECTRIC	UNITS	DAILY Total	7-DAY AVG	% DIFF	COST
METER X122-0	KWH	948	927	2.2 %	$62.83
METER X122-1	KWH	828	900	8.0 %	$54.89
WEATHER - TEMP	UNITS	DAILY Avg	7-DAY AVG	% DIFF	COST
AMBIENT AIR TEMPERATURE	DEGF	63.64	58.02	9.7 %	

Figure 4. Utility report

Hypertext Markup Language (HTML)

Using well-proven and accepted web page creation tools is the way to insure reliability in a web page. There are many ways to present data using a web page, but not all browsers interpret those methods the same. Using basic hypertext markup language (HTML) and following with simple enhancements will insure reliable results. You create a file and put special character sequences called HTML elements into your file. These elements identify the structural parts of your document. When a web browser displays the file, it will display the content, but not the characters that make up the structure. (2) Below is a very simple HTML example with HTML element tags.

```
<html>
<head>
<title>
A Small Hello
</title>
</head>
<body>
<h1>Hi</h1>
<p>This is very minimal “hello world” HTML document.</p>
</body>
</html>
```

This is what the HTML page above looks like when displayed in a browser:

Hi

This is a very minimal “hello world” HTML document.

One way to learn HTML is to go through the tutorial on the W3Schools web site at http://www.w3schools.com/html. Another way is to enroll in a web-page design class at a local community college.

Extensible Hypertext Markup Language (XHTML)

Extensible hypertext markup language (XHTML), the latest version of HTML, provides even more reliability with its strict rules. XHTML gives you the opportunity to write “well-formed” documents that work in all browsers and that are backward browser compatible. Developers can validate XHTML pages for correctness using tools available on-line. Below is an example of a very simple XHTML file.

```
<?xml version=”1.0" encoding=”iso-8859-1"?>
<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Transitional//
    EN” “DTD/xhtml1-transitional.dtd”>
<html xml:lang=”en” lang=”en” xmlns=”http://www.w3.org/1999/
    xhtml”> <head>
        <title>Hello World</title>
    </head>
    <body>
        <p>My first Web page</p>
    </body>
</html>
```

The XHTML page above looks like this when displayed in a browser: (3)

My first Web page.

You might also consider completing the XHTML tutorial on the W3Schools web site at *http://www.w3schools.com/xhtml*.

Cascading Style Sheets (CSS)

The World Wide Web Consortium (W3C), the non-profit standard-setting consortium responsible for standardizing HTML and XHTML, created styles in addition to HTML 4.0. Styles define how to display HTML elements. Using cascading style sheets (CSS) can minimize the amount of HTML code needed to display data and therefore make the pages less complex, faster loading, and easier to maintain. Below is a very simple CSS example. (4)

```
<head>
    <style>
        b {color: red}
        i {color: blue}
    </style>
</head>
<body>
This is how to use <b>CSS</b> to brighten up your <i>World</i>!
</body>
```

The page looks like this when displayed in a browser: (5)

This is how to use CSS to brighten up your *World*!

You can find a CSS tutorial on the W3Schools web site at *http://www.w3schools.com/css*.

Dynamic HTML (DHTML)

HTML 4.0 also introduced the document object model (DOM). The DOM gives us access to every element in the document. By using HTML, CSS and JavaScript, web pages appear to be "dynamic." Dynamic HTML (DHTML) is not a standard defined by the World Wide Web Consortium (W3C). DHTML is a combination of technologies web developers use to control the display and position of HTML elements in a browser window. Below is a simple example of DHTML. (6)

```
<html>
<body>
<h1 id="header">My header</h1>
<script type="text/javascript">
```

```
header.style.color=”red”
</script>
</body>
</html>
```

The page looks like this in the browser:

My header

The DHTML tutorial on the W3Schools web site is at *http://www.w3schools.com/dhtml*.

Sorting, Filtering and Graphing

Qualifying the data with headings provides the opportunity to add links to sorting, filtering, and graphing routines. Links in the display provide plenty of robust functionality to the user.

The display below shows headings with links to sorting routines:

Select this link to sort by utility

Facility	Utility	Consumption	Units	Cost
BUILDING-1	ELECTRIC	6480.33	KWH/DAY	$ 18144.92
BUILDING-1	WATER	6291.14	GALLONS/DAY	$ 4592.53
BUILDING-2	ELECTRIC	6938.27	KWH/DAY	$ 19427.16
BUILDING-2	WATER	6653.16	GALLONS/DAY	$ 4856.81

The display below shows values sorted by utility and links to filter routines:

Select this link to filter for Building 1

Facility	Utility	Consumption	Units	Cost
BUILDING-1	ELECTRIC	6480.33	KWH/DAY	$ 18144.92
BUILDING-2	ELECTRIC	6938.27	KWH/DAY	$ 19427.16
BUILDING-1	WATER	6291.14	GALLONS/DAY	$ 4592.53
BUILDING-2	WATER	6653.16	GALLONS/DAY	$ 4856.81

The display below shows values filtered for Building 1 and links to graphing routines:

Select this link to graph

Facility	Utility	Consumption	Units	Cost
BUILDING-1	ELECTRIC	6480.33	KWH/DAY	$ 18144.92
BUILDING-1	WATER	6291.14	GALLONS/DAY	$ 4592.53

Figure 5 below shows a graph of electric consumption per day for Building 1 over the last 30 days.

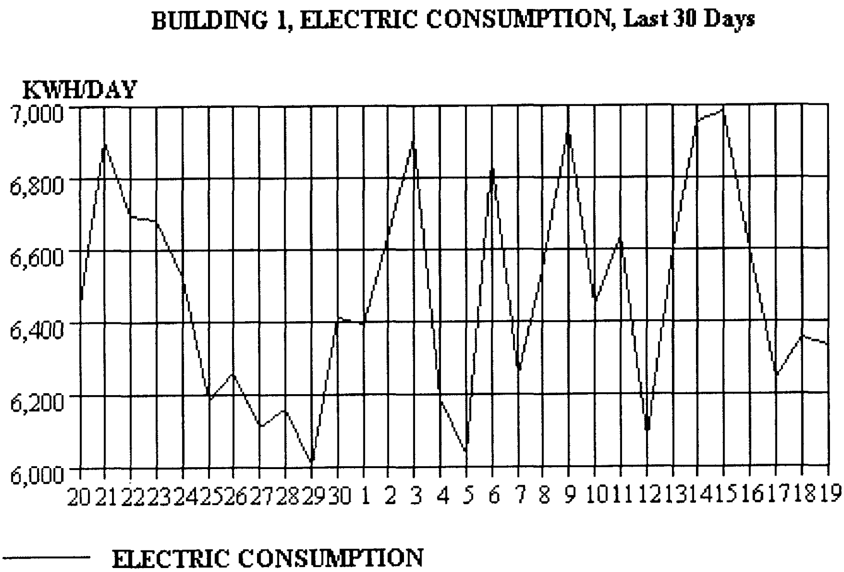


Figure 5. Electric consumption over the last 30 days

Extensible Markup Language (XML)

Data stored in the form of extensible markup language (XML) allows for quick access to data attributes and the ability to customize the display based on those attributes using CSS.

Appendix A illustrates an example of an XML file with its schema included at the top of the file.

Appendix B illustrates an example of a CSS file:
Below is what the two files above look like in a browser: (7)

AH 101 ELECTRIC KWH
455.34 2001-09-13T11:30:06
578.29 1002-09-13T11:45:04
479.30 2001-09-13T12:00:05
AH 102 ELECTRIC KWH
347.59 2001-09-13T11:30:06
498.60.2001-09-13T11:45:04
444.57 2001-09-13T12:00:05

You can see how these files work together by visiting the UtilityReporting.com web site.

- The XML file is at *http://utilityreporting.com/utilitydata.xml*
- The CSS file is at *http://utilityreporting.com/utilitydata.css*
- The combination of the two files is at *http://utilityreporting.com/utilitydata2.xml*

You might also want to visit the XML tutorial on the W3Schools web site at *http://www.w3schools.com/xml*.

Server-side Technologies

Common gateway interface (CGI) provides a convenient method of running the intensive server operations required for utility data analysis using a web browser. The first generation of web server-side technologies was CGI. Many developers use the PERL programming language to create CGI applications. They also use nearly any higher level language. This technology employs server-side applications to complete a predefined task when the client makes a request. The following line from the address field of a web browser runs a server application to query a database according to the parameters and return the results to the browser.

http://localhost/scripts/foxweb.exe/A1@view?sortby=cost~date=12/08/2002~complex=A~building=1~level=building~

The next generation of server-side technologies is server-side scripts such as Active Server Pages, Java Servlet/Java Server Pages, PHP, and others. These technologies allow the server to generate a thread to handle a client's request. The thread is light-weighted and consumes fewer resources than the application generated by CGI. On the other hand, server-side scripting languages often provide only a subset of the capabilities of full programming languages. (6)

Client-side Technologies

Unlike server-side technologies, client-side technologies execute tasks at the client end. There are a number of popular client-side technologies available to implement extra functionality on top of HTML. Three of the major client-side technologies are client-side scripts, Java applets and ActiveX controls. Client-side scripts such as JavaScript, VBScript, and Jscript are a collection of commands interpreted, rather than compiled, by the web browser. A Java applet is a Java program downloaded from a web server into client browsers and runs inside a Java virtual machine (JVM) that is integrated into the web browser. The term ActiveX generally refers to ActiveX controls that are software modules based on Microsoft's component object model (COM) architecture. The power of ActiveX controls in web development is that any ActiveX-compliant web browser can download the controls by clicking on its link. This turns web pages into virtual software pages that can perform operations just like any desktop application. (8)

Below is an example of JavaScript:

```
<html>
<body>
<script type=”text/javascript”>
document.write(“Hello World!”)
</script>
</body>
</html>
```

This is what the above page looks like in a web browser:

Hello World!

Configuration Tables

Configuration tables allow for color definitions, text lookup and runtime adjustment of critical constants.

Table 2. Configuration table

LANGUAGE	*NAME*	*VALUE*
English	Title	Energy Information System
English	Facility	Facility
English	Utility	Utility
English	Consumption	Consumption
English	Units	Units
English	Cost	Cost
English	Background color	#A5D1D1
English	Text color	#000000
English	Link color	#0000FF
French	Title	Système D'Information D'Énergie
French	Facility	Service
French	Utility	Utilité
French	Consumption	Consommation
French	Units	Unités
French	Cost	Coût
French	Background color	#A5D1D1
French	Text color	#000000
French	Link color	#0000FF

All of this leads to a more robust look and feel. Links allow the user to change the display with a single selection. HTML and XHTML provide reliable presentation of the data in the web browser window. XML combined with CSS provides a direct association between the raw data attributes and the presentation of the data. JavaScript manipulates elements of the document object model (DOM). Configuration tables allow developers to make major changes in a relatively short time with very little effort.

CONCLUSION

Facilities have an obvious need to track utility use to conserve and troubleshoot equipment problems. The more people are involved in the process, the easier the task. Good web page design and web publishing technologies make successful utility tracking a bright reality.

Making good use of basic HTML or XHTML along with XML combined with CSS can produce plenty of good reports. DHTML used sparingly can make the pages user-friendlier. CGI scripts are a simple and effective way to connect web pages with server applications. Organizing utility data on a web page in tabular form with links to sort, filter and graph provides an informative view toward utility monitoring. This allows people to make informed decisions in a timely manner concerning utility use and conservation.

An excellent source for learning these web publishing technologies is W3schools.com at *http://www.w3schools.com*. There you will find free web-building tutorials from basic HTML and XHTML tutorials to advanced XML. A number of books are available on the subject. A good place to start is the book *10 Minute Guide to HTML* by Tim Evans, published by QUE Corporation. Local community colleges might offer classes as well. The next article in this series will discuss in detail web publishing technologies required for good utility data web page design. Then a third article will describe how to use those technologies to present utility data in way to make maximum use of its unique characteristics described above.

References

(1) 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.

(2) December, John, "Creating Web Documents" [*article on-line*] (January 20, 2003, accessed 12 February 2003); available from http://www.december.com/html/tutor/hello.html; Internet.

(3) UK Web Design, Internet Marketing, Usability Testing; "XHTML Web Design for Beginners: Hello World" [*article on-line*] (2003, accessed 12 February 2003); available from http://www.miswebdesign.com/resources/articles/web-design-xhtml-1-2.html; Internet.

(4) "Introduction to CSS" [*article on-line*] (2003, accessed 27 February 2003); available from http://www.w3schools.com/css/css_intro.asp; Internet.

(5) Wisman, Raymond F., "Cascading Style Sheets"[*article on-line*] (December 13, 2002, accessed 12 February 2003); available from http://homepages.ius.edu/RWISMAN/A346/html/CSS.htm; Internet.

(6) "Introduction to DHTML" [article on-line] (2003, accessed 27 February 2003); available from http://www.w3schools.com/dhtml/dhtml_intro.asp; Internet.

(7) Green, David C., "UtilityReporting.com"[article on-line] (January 12, 2003, accessed 12 February 2003); available from http://utilityreporting.com/utilitydata.xml; Internet.

(8) Fanqzing Li, Lavelle A.A. Freeman, "Using Client-side Technologies to Develop Web-based Applications for Power Distribution Analysis," Distributech 2003 Conference, February 4-6, 2003, Las Vegas, Nevada.

ABOUT THE AUTHORS

David C. 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)

Paul J. 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)

Appendix A


```
<?xml version=”1.0" ?>
<UtilityData>
<Schema name=”Schema” xmlns=”urn:schemas-microsoft-com:xml-data”
xmlns:dt=”urn:schemas-microsoft-com:datatypes”>
<ElementType name=”AirHandlers” >
<element type=”AirHandler”/>
</ElementType>
<ElementType name=”AirHandler” >
<element type=”AirHandlerNo”/>
<element type=”Utility”/>
<element type=”Units”/>
<element type=”Readings”/>
</ElementType>
<ElementType name=”AirHandlerNo” dt:type=”string”>
<AttributeType name=”type”/>
<attribute type=”type” default=”string”/>
<AttributeType name=”size”/>
<attribute type=”size” default=”8"/>
</ElementType>
<ElementType name=”Utility” dt:type=”string”>
<AttributeType name=”type”/>
<attribute type=”type” default=”string”/>
<AttributeType name=”size”/>
<attribute type=”size” default=”30"/>
</ElementType>
<ElementType name=”Units” dt:type=”string”>
<AttributeType name=”type”/>
<attribute type=”type” default=”string”/>
<AttributeType name=”size”/>
<attribute type=”size” default=”10"/>
</ElementType>
<ElementType name=”Readings” >
<element type=”Reading”/>
</ElementType>
<ElementType name=”Reading” >
<element type=”Consumption”/>
<element type=”DateRecorded”/>
</ElementType>
<ElementType name=”Consumption” dt:type=”float”>
<AttributeType name=”type”/>
<attribute type=”type” default=”float”/>
```

```
<AttributeType name=”size”/>
<attribute type=”size” default=”15"/>
<AttributeType name=”precision”/>
<attribute type=”precision” default=”2"/>
</ElementType>
<ElementType name=”DateRecorded” dt:type=”datetime”>
<AttributeType name=”type”/>
<attribute type=”type” default=”datetime”/>
<AttributeType name=”size”/>
<attribute type=”size” default=”8"/>
</ElementType>
</Schema>
<AirHandlers xmlns=”x-schema:#Schema”>
<AirHandler>
<AirHandlerNo>AH 101</AirHandlerNo>
<Utility>ELECTRIC</Utility >
<Units>KWH</Units >
<Readings>
<Reading>
<Consumption>455.34</Consumption>
<DateRecorded>2001-09-13T11:30:06</DateRecorded>
</Reading>
<Reading>
<Consumption>578.29</Consumption>
<DateRecorded>2001-09-13T11:45:04</DateRecorded>
</Reading>
<Reading>
<Consumption>479.30</Consumption>
<DateRecorded>2001-09-13T12:00:05</DateRecorded>
</Reading>
</Readings>
</AirHandler>
<AirHandler>
<AirHandlerNo>AH 102</AirHandlerNo>
<Utility>ELECTRIC</Utility>
<Units >KWH</Units>
<Readings>
<Reading>
<Consumption >347.59</Consumption>
<DateRecorded >2001-09-13T11:30:06</DateRecorded>
</Reading>
<Reading>
```

```
<Consumption >498.60</Consumption>
<DateRecorded >2001-09-13T11:45:04</DateRecorded>
</Reading>
<Reading>
<Consumption >444.57</Consumption>
<DateRecorded >2001-09-13T12:00:05</DateRecorded>
</Reading>
</Readings>
</AirHandler >
</AirHandlers>
</UtilityData>
```


Appendix B

```
UtilityData
{
background-color: #ffffff;
width: 100%;
}
AirHandlers
{
display: block;
margin-bottom: 12pt;
margin-left: 0;
}
AirHandler
{
color: #FF0000;
font-size: 10pt;
}
Reading
{
Display: block;
color: #000000;
font-size: 10pt;
margin-left: 10pt;
}
```