Opendiem Training

Exercise 11

Opendiem-TRN-0011

Revision	5.0.0		
Status	Initials	Date	Comment
Released	RAC	11/18/2010	Updated format and content

Exercise 11 – Opendiem Data Logger Service

Introduction

The Opendiem Data Logger service logs Opendiem data based on specified conditions or interval. The logged data can be retrieved using a standard Web Browser and displayed as a Graph, Report or as XML data.

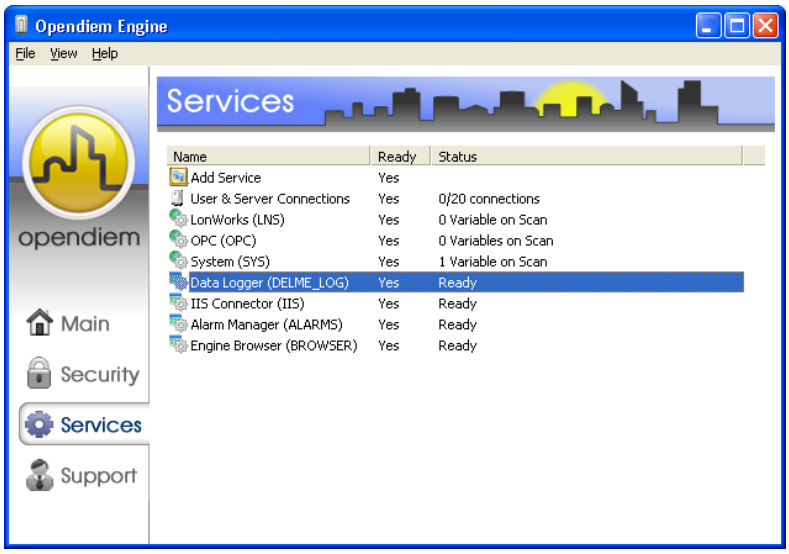
Objective

In this exercise you will use the Opendiem Data Logger service to log variables of various types using the 'drag & drop' interface. You will learn the various logging options and also how to export data to a spreadsheet format for off-line editing. You will also display the logged data in a standard Web Browser in graphical and tabular formats.

Starting Opendiem Data Logger Service

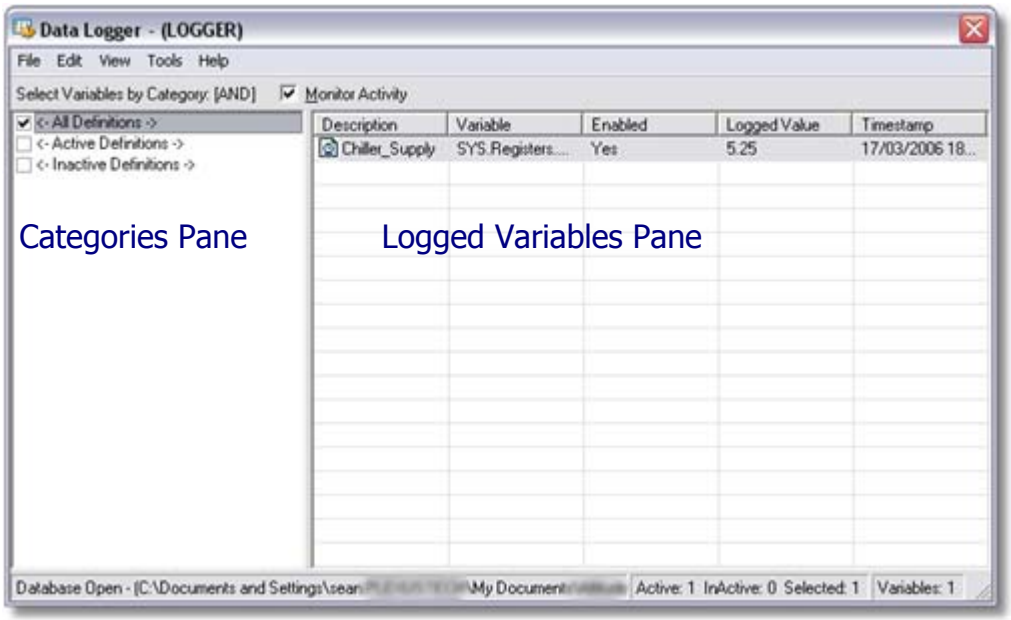
Exercise Instructions

Ensure that Opendiem Engine is running and if necessary restart it.



From the Services screen of Opendiem Engine check that the Opendiem Data Logger service is ready, if necessary add the service from the Client services.

Double click on the service to configure it. The following screen appears:



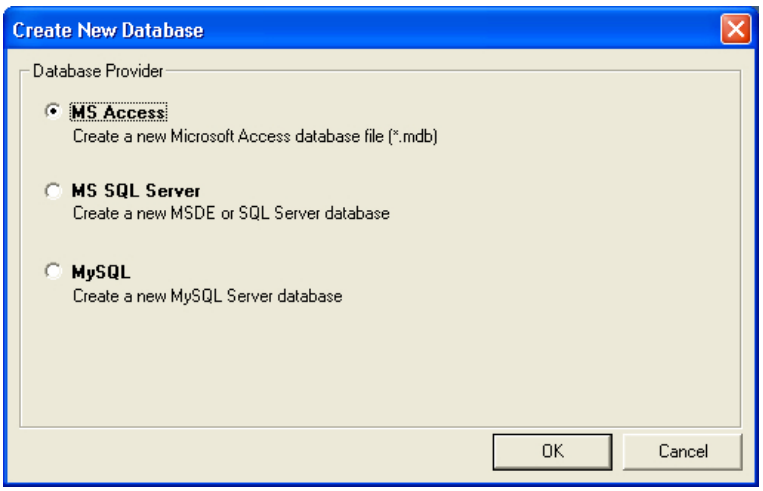
On starting the Data Logger service for the first time there may be no database configured or the database may be closed. The Data Logger service has ODBC drivers built in which allow it to operate with ODBC compliant databases such as SQL. Operation with these database formats is

beyond the scope of this training class and we will therefore be operating on an Access database (.mdb).

To create a new database:

Select **File | New Database...**

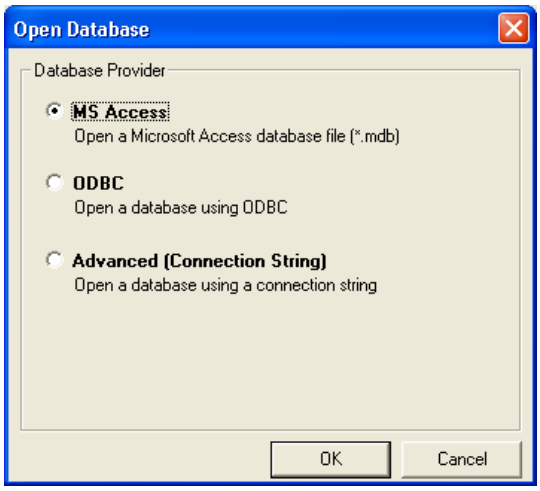
Select MS Access database.



Browse to '`C:\Documents and Settings\myuser\My Documents\Opendiem\myproject`' and enter a name for the database e.g. '`logtraining.mdb`'.
Select **Save**.

To open an existing database:

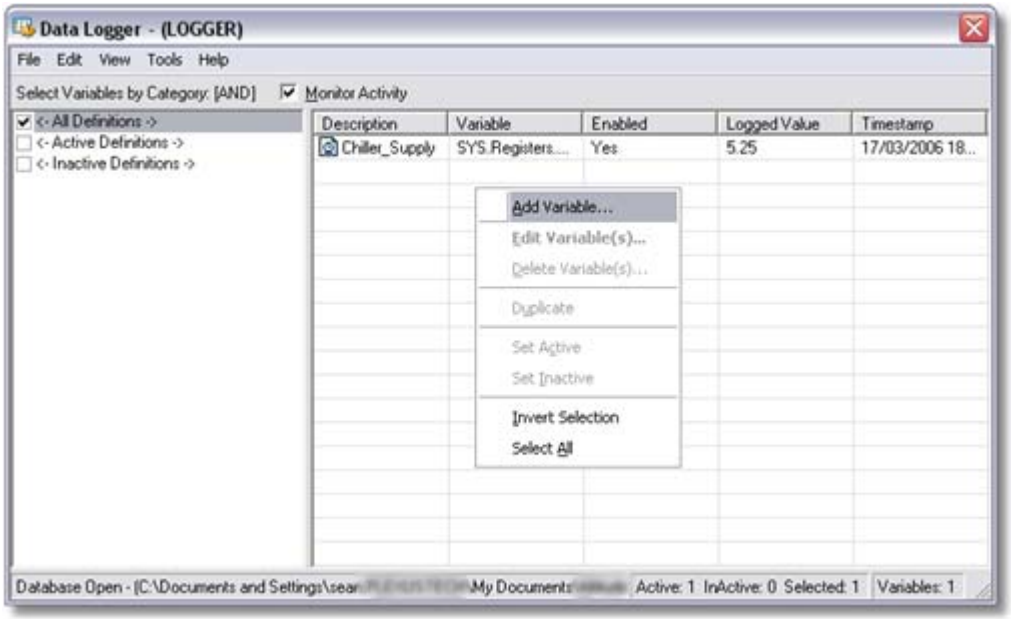
Select **File | Open Database...**
Select the database type. For this training class select a standard database file (*.mdb). Other database types are beyond the scope of this training class.



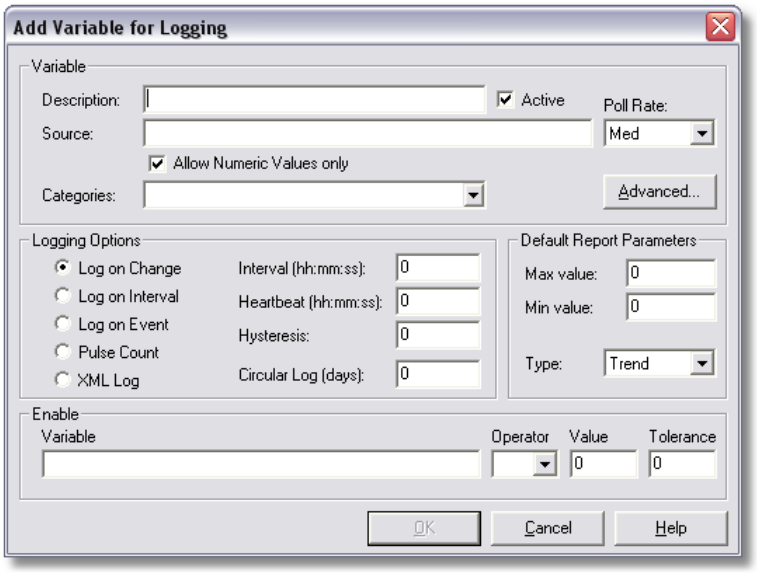
Browse to the database and select **Open**.

Any Data Logs configured in the database appear on the screen panes.

Add a Data Log Variable by right clicking in the Logged Variables Pane and selecting **Add Variable...**.



A Data Logging dialogue will appear as shown below.



The default logging type is **Log On Change**. There are currently five logging types configured in the Data Logger service, Log On Change, Log On Interval, Log On Event, Pulse Count and XML Log. Each of these types is described below.

Log On Change	The Log On Change type logs a variable value when the variable changes by more than the hysteresis value. Optional interval and heartbeat parameters control the logging interval.
Log On Interval	The Log on Interval type logs a variable value at the specified interval.
Log On Event	The Log on Event type logs a variable value when an event from another Opendiem service is received e.g. from the Scheduler or Alarm service.
Pulse Count	The pulse count log counts pulses from external sources such as metering pulses. This option can realistically only be used on bound connections to an LNS server. The interval parameter selects the totalization period before adding the log to the database.

XML Log — This produces a data log in one "dump" instead of over a period of time. This is designed for use with devices like the Trend Controller which stores data logs internally. The XML data log will download the log from the Trend Controller and enter the data into the database.

This option is used with the Interval property and has its own property; Min Request Time which specifies the minimum time between downloads.

The default settings for the XML Log are available on the Logger options dialog.

Configuring a Data Log

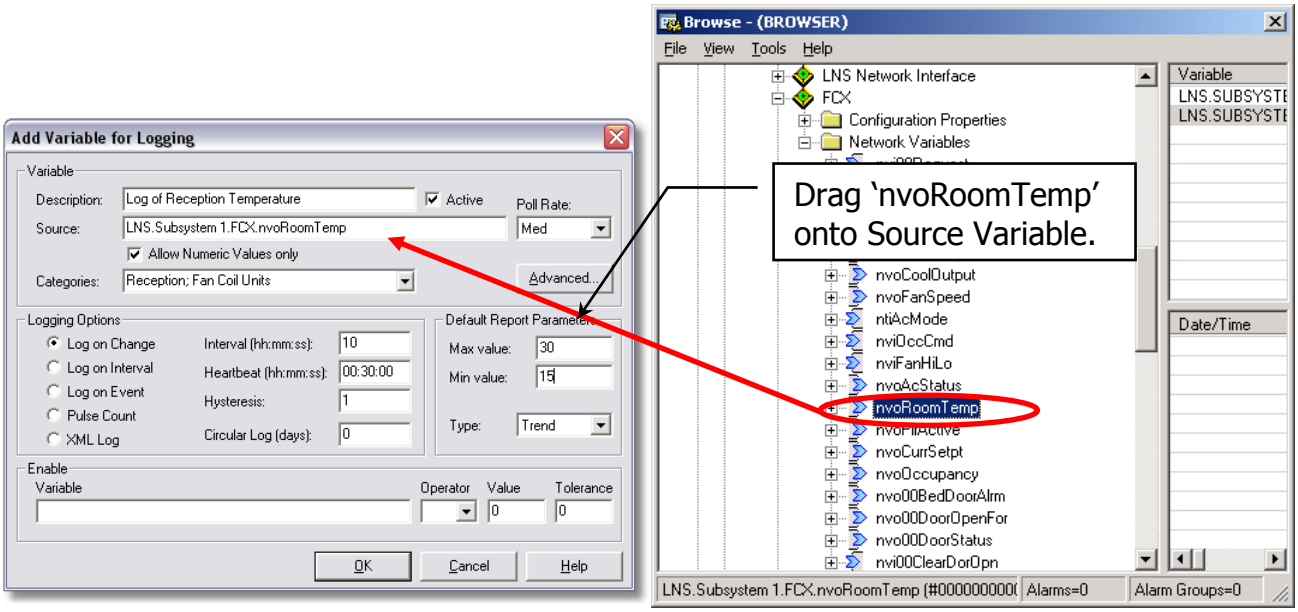
In this exercise we will set up a Log on Change variable log as it is the most comprehensive, other Data Log types are configured in the same way.

Data Log Options

Description. — The Description field is present to document the Logged variable.

Allow Numeric values only — When checked, only numeric values will be logged. This prevents the log being filled with enumerated text data which is impossible to plot.

Source Variable. — The Source variable is the Tag name of the variable to be logged. Either enter the full Tag name or drag & drop the variable from the Opendiem Project Browser as shown below.



Categories.	Each Logged variable may belong to one or more categories. Categories allow multiple data logs to be selected for editing and also for events such as Log on Event log types.
Logging Options	The Logging options are specific to the log type such as interval, hysteresis etc.
Circular Log	A log may grow to the size allowed for the database or a circular log can be setup with a duration specified in days. Circular logs replace the oldest data in the log with new data (0=infinite, not recommended).

Log Enable

The Enable field contains details on an optional variable that may enable & disable this log.

Variable	This field contains the Tag reference for the enabling variable and may be dragged from the Project Explorer as shown previously.
Operator	The mathematical operator that will be used then comparing the value Opendiem Variable to the specified enable value.
Value	The value that will be compared to the value of the Opendiem Variable when determining whether to enable this log.

Tolerance	The tolerance that will be applied to the comparison if the operator that is being used is equals (=) or not equals (<>).

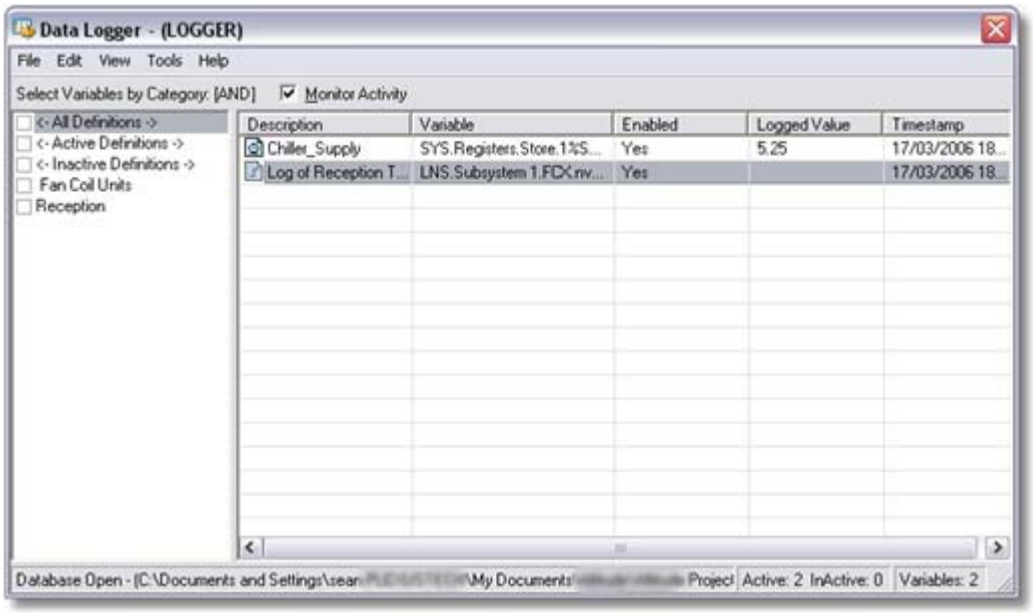
Default reporting Parameters

In this section, the default reporting parameters are specified for when the data is presented in a chart within the Web Browser using the OpendiemGraph Java client.

Maximum Value	This field contains the maximum Y axis value for the chart.
Minimum Value	This field contains the minimum Y axis value for the chart.
Type	This drop down list specifies the default graph type and can be Trend, Event or Bar Chart. An Event graph type will display a narrow horizontal line for data with a value of zero and a thicker horizontal line for non-zero data.

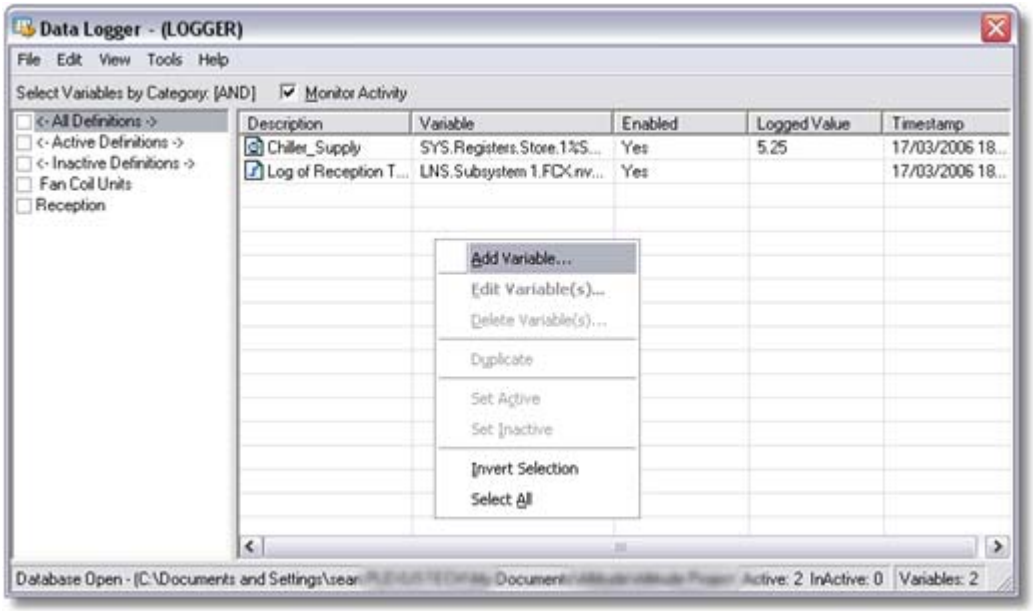
Note: If both the Maximum and Minimum Values are set to zero the graph will auto scale to fit all of the data.

Select **Save** and the log parameters will be saved to the database and the screen similar to the one below will appear. This screen gives an overview of the Data Logger including all of the categories, the description of each log, the variable Tag, the enabled status and the last value written to the database.



Right clicking on the individual data logs reveals a context menu which allows you to add, edit, delete, duplicate, enable, disable, invert selections and select all logs.

Experiment by adding new data logs and categories for your project.



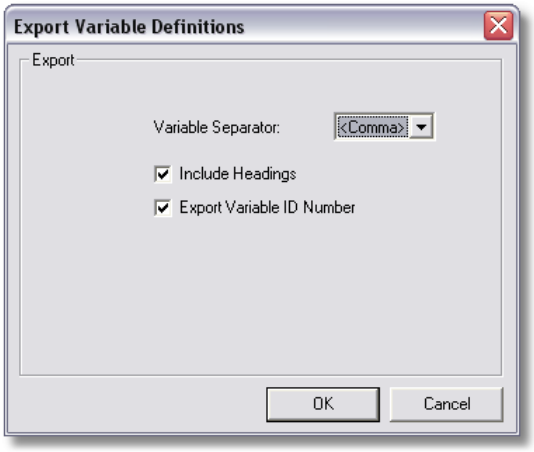
Checking the Database

The status of the Logger database can be inspected by selecting **File | Properties**. This screen shows the database type, the path and name of the Data Logger database and the total size of the database.

Importing & Exporting the Database

For large systems, it is convenient to export the database as a CSV type file which can be edited in a spreadsheet such as Excel. In this way many log parameters can be set extremely quickly and then re-imported to the Opendiem Database. The export also acts as a convenient backup of the settings.

From the main menu select **File | Export Configuration**, **Export to File...** The screen below will appear. Select the preferred separator, tab or comma and click **OK**.



You will be prompted for a filename, choose a name and select **Save**. A text file suitable for reading into a spreadsheet will be created. If there is no spreadsheet program installed on your PC you can inspect the file using notepad.

The file is imported back into Opendiem using **File | Import Configuration**, **Import from File...**
Remember to select the correct delimiter before clicking **Open**.

Viewing the Logged Data

Data stored by the data Logger service can be presented within a Web Browser in either a Graph format or as a table. Opendiem exports this data as industry standard XML which is formatted using server side XSLT style sheets to display the data any way required.

Background information on XML & XSLT

XML is the Extensible Markup Language. It is designed to improve the functionality of the Web by providing more flexible and adaptable information identification.

It is called extensible because it is not a fixed format like HTML (a single, predefined markup language). Instead, XML is actually a `metalanguage' —a language for describing other languages—which lets you design your own customized markup languages for limitless different types of documents. XML can do this because it's written in SGML, the international standard metalanguage for text markup systems (ISO 8879).

In HTML, default styling was built into the browsers because the tagset of HTML was predefined and hardwired into browsers. In XML, where you can define your own tagset, browsers cannot possibly be expected to guess or know in advance what names are going to be used and what they will mean, so a stylesheet is required to display formatted text.

Browsers which read XML usually accept and use a CSS stylesheet at a minimum, but it is also possible to use the more powerful XSLT stylesheet language to transform XML into HTML—which browsers already know how to display (HTML can still use a CSS stylesheet). In this way all of the document management benefits of using XML are obtained, but you don't have to worry about users needing XML smarts in their browsers.

XSLT is an XML document processing language that uses source code that happens to be written in XML. An XSLT document declares a set of rules for an XSLT processor to use when interpreting the contents of an XML document. These rules tell the XSLT processor how to generate a new XML-like data structure and how that data should be emitted—as an XML document, as an HTML document, as plain text, or perhaps in some other format.

This transformation can be done either inside the browser, or by the Opendiem server before the file is sent. Transformation in the browser offloads the processing from the server, but may introduce browser dependencies, leading to some of your users being excluded. Transformation on the Opendiem server makes the process browser-independent, but places a heavier processing load on the server.

As with any system where files can be viewed at random by arbitrary users, the author cannot know what resources (such as fonts) are on the user's system, so the same care is needed as with HTML using fonts.

Older web browsers may not handle XSLT inside the browser (MSIE5.5 needs some post-installation surgery to remove the obsolete WD-xsl and replace it with the current XSL-Transform processor; MSIE6+ and Mozilla work as delivered).

Benefits of XML

XML key features and applications:

1. XML is not a markup language. XML is a `meta language' , that is, it's a language that lets you define your own markup languages (see definition).
2. XML *is* a markup language (*not* a programming language). XML is data: it does not `do' anything, it has things done to it.
3. XML is non-proprietary: your data cannot be held hostage by someone else.
4. XML allows multi-purposing of your data.
5. Well-designed XML applications like Opendiem separate `content' from `presentation' . It should describe what something *is* rather what something *looks like* (the exception being data content which never gets presented to humans).

XML is currently an industry 'buzzword', but saying `the data is in XML' is a relatively useless statement, similar to saying `the book is in a natural language' . To be useful, the former needs to specify `we have used XML to define our own markup language' (and say what it is), similar to specifying `the book is in French' .

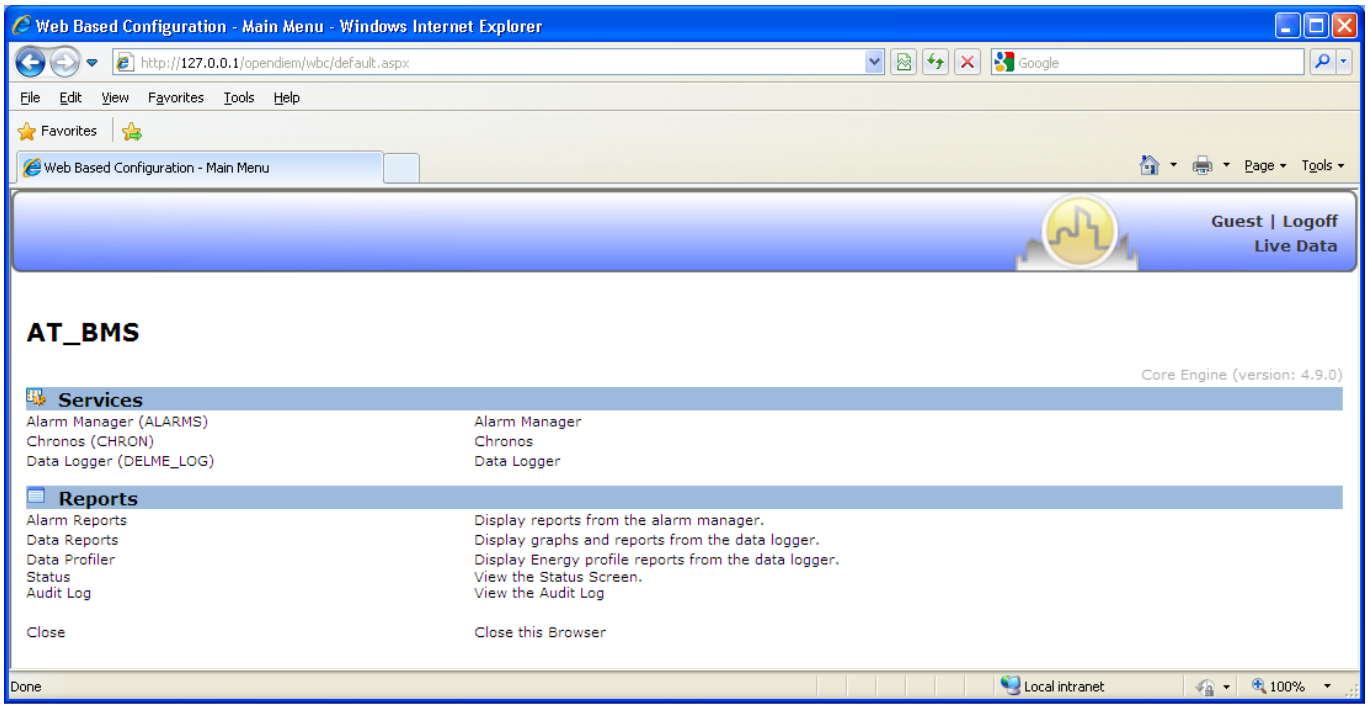
An example of multipurposing and separation can be applied to a pharmaceutical company. For example, they have a large base of data on a particular drug that they need to publish as:

- reports to the FDA;
- drug information for publishers of drug directories/catalogs;
- `prescribe me!' brochures to send to doctors;
- leaflets that are inserted into the boxes;
- labels on the bottles;
- pages of fine print;
- instructions to the patient that the local pharmacist prints out;
- etc.

Without separation of content and presentation, it would be necessary to maintain essentially identical information in 20 places. If one of these places was to be missed then there would inevitably be big problems. With XML, one set of carefully validated information can be maintained, and 20 programs could be written to extract and format it for each application. When correctly applied the same 20 programs can be used for all of the hundreds of drugs that are made.

XML and style sheets are beyond the scope of this training class, but we can view the data by starting a Web Browser.

Start a browser connection to Opendiem. In the address bar enter http://127.0.0.1/Opendiem/wbc/wbc.html. Substitute the IP address of your own PC if necessary. WBC is the web based configuration. The screen below will appear.



Select **Data Reports**

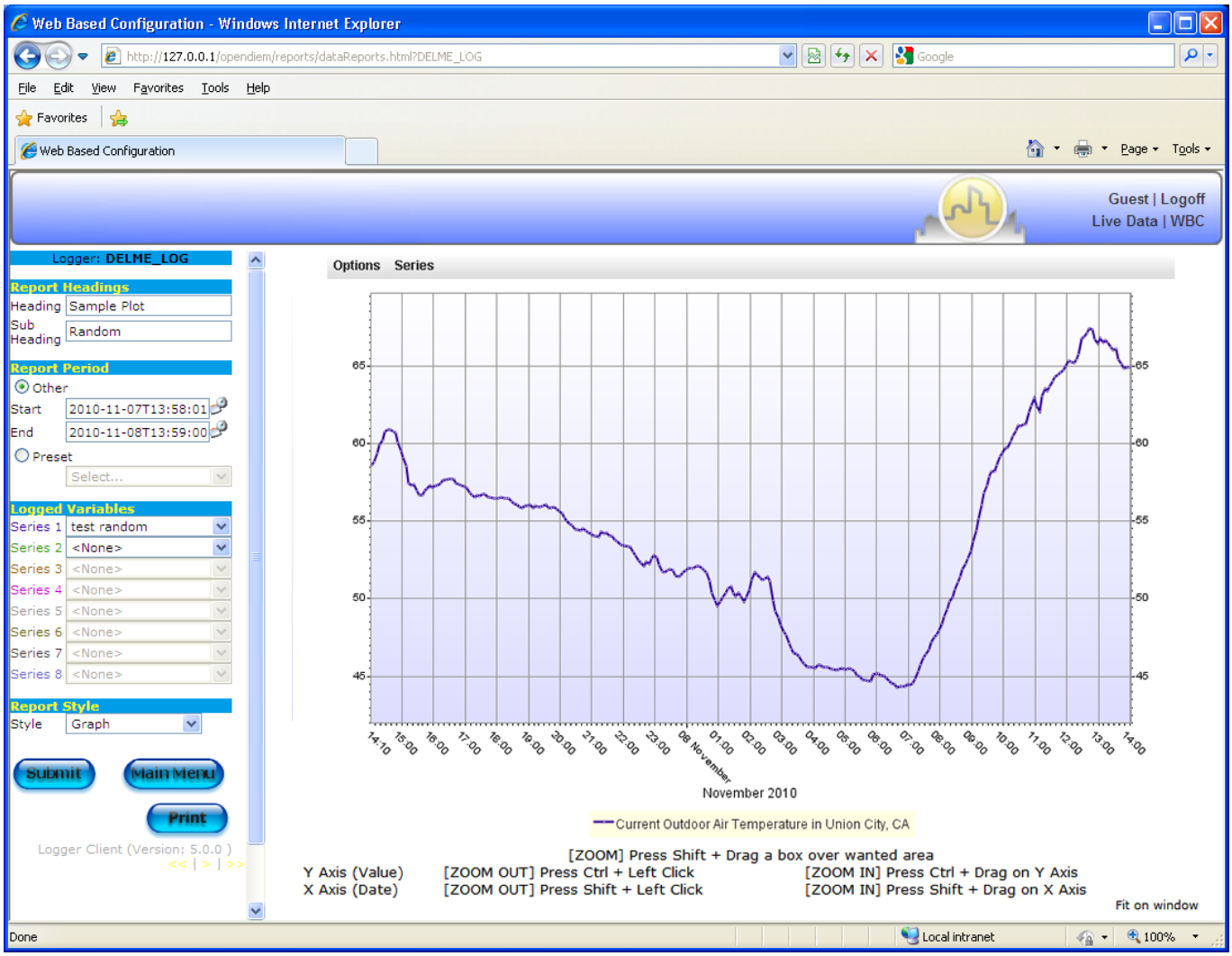
A screen similar to the one below will appear.

Select start and end dates from the pop-up calendars which appear when you click on the calendar icons next to the start and end boxes. If necessary edit the times of the start and finish. The Date & Time is set in W3C web standard format. Enter some text to label your data in the heading & sub-heading boxes.

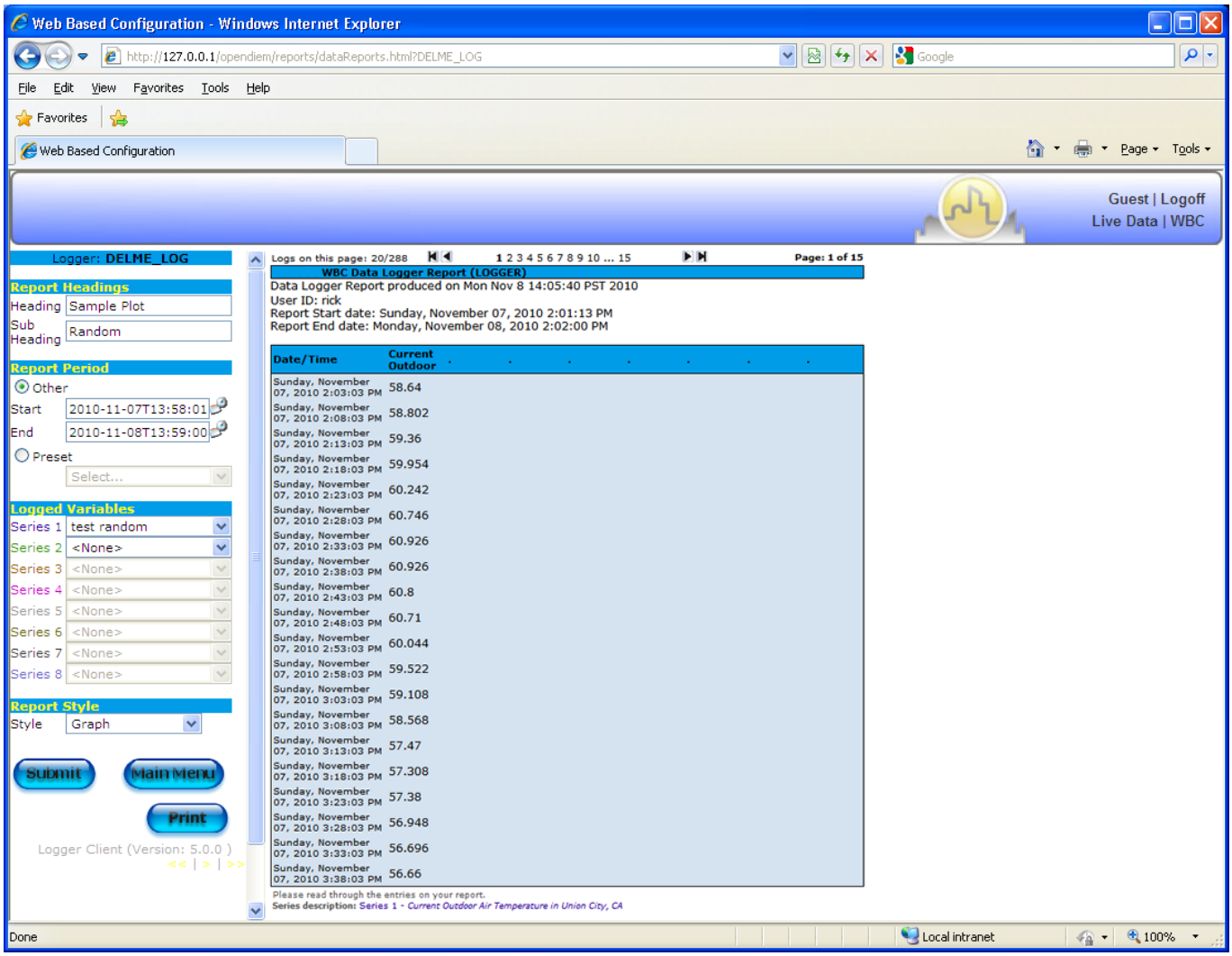
In the drop down lists any logs that you set up in the Data Logging service will automatically appear.

Select up to eight series to display on the same chart, select Graph as the Report Style and click **Submit**.

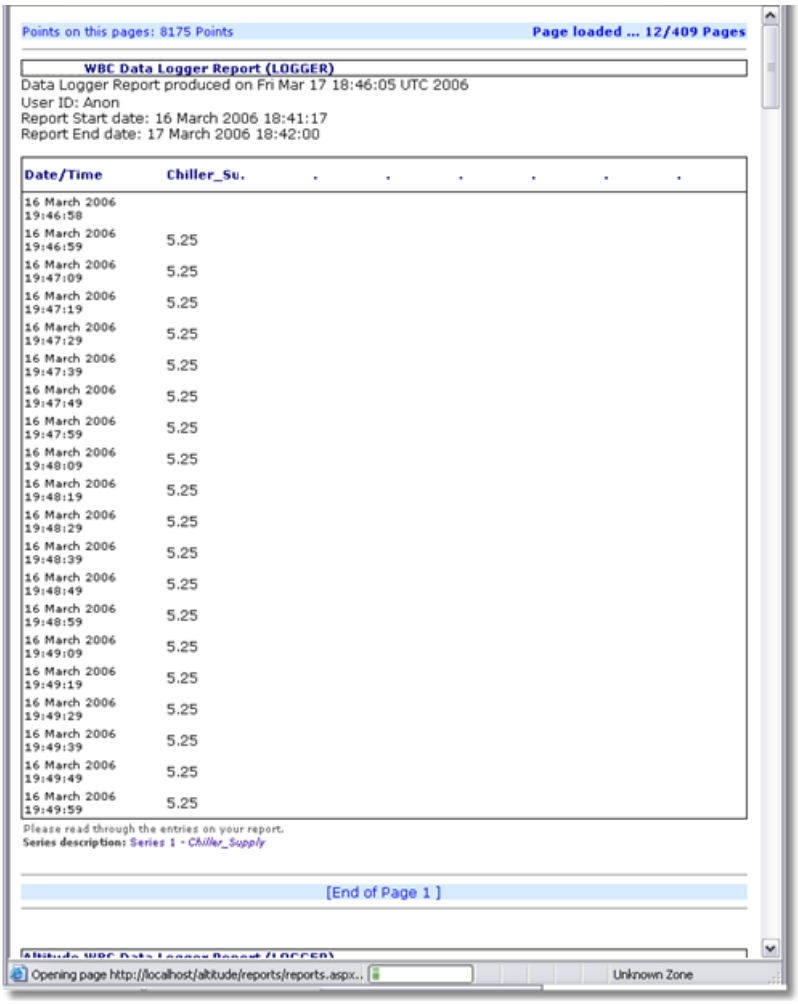
The browser will post a request to Opendiem and the data will be collected from the database. This data will then be formatted to XML and a style sheet on the server is used to format the data. The graph will appear as shown below.



In the **Style** drop down list select **Paged Report, 100 per page** and then select **Submit**. This time the data has a different XSLT style sheet applied on the Opendiem Server and the information is returned as a report style as shown below.



Click on **Print** and the report will be formatted so that it will easily print as a report as shown below.



The Opendiem Web Based Configuration (WBC) is provided as an example of what can be achieved with the WBC. All of the WBC files are provided as open format and may therefore be modified as required.

End of Exercise 11

In this exercise we have used the Opendiem Data Logger service to set up some sample data logs. We have also discovered how to export and import the configuration database for rapid editing in a spreadsheet.

You have now also displayed the logged data in a Web Browser as a Graph and also as a report using XML.

Notes:

Version 5.00 November 2010

All Rights Reserved. Copyright © 2010 Building Clouds. Information contained in this document regarding device applications is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Building Clouds, Inc. with respect to accuracy or use of such information or infringement of patents arising from such use or otherwise. Use of Building Clouds products as critical components in life support systems is not authorized except with expressly written approval by Building Clouds. No Licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

All Trademarks mentioned herein are the property of their respective companies

Building Clouds
3229 Whipple Road
Union City, CA 94587

Email: support@buildingclouds.com
http://www.buildingclouds.com