# Table of Contents

1 Introduction ........................................................................................................................................... 1
   1.1 Supported Sensor Boards and Mote Platforms ................................................................. 1
   1.2 Supported Mote Software Applications ........................................................................ 2

2 Installation .......................................................................................................................................... 6
   2.1 Supported PC Platforms and Operating Systems .......................................................... 6
   2.2 Additional Software Requirements ................................................................................ 6
   2.3 Installation Steps ............................................................................................................... 7

3 Application Quick Start ................................................................................................................. 11
   3.1 Connecting to the Demo Database on Your PC ............................................................. 11
   3.2 Connecting to an Active Sensor Network Using an MIB510 or MIB600 .................... 12
   3.3 Connecting to a Database (Local or Remote) Server or a Stargate ............................. 14
   3.4 Auto-discovery of Nodes ............................................................................................... 15

4 User Interface Functional Description ........................................................................................ 16
   4.1 MOTE-VIEW at a Glance ............................................................................................... 16
   4.2 Node List ....................................................................................................................... 16
   4.3 Visualization Tabs ........................................................................................................ 16

5 Menus ............................................................................................................................................. 28
   5.1 FileConnect .................................................................................................................. 28
   5.2 Tools ............................................................................................................................ 29
   5.3 Units ............................................................................................................................ 37
   5.4 Windows ...................................................................................................................... 38
   5.5 Help ............................................................................................................................. 38
   5.6 Toolbar Menu ............................................................................................................... 38
   5.7 Dialogs ........................................................................................................................ 39
   5.8 Server Messages ........................................................................................................... 41

6 Server Administration .................................................................................................................... 42
   6.1 General configuration .................................................................................................... 42
   6.2 XServe Logging .......................................................................................................... 42

7 Database Administration ................................................................................................................. 44
   7.1 PostgreSQL .................................................................................................................. 44
   7.2 SQL ............................................................................................................................. 44
   7.3 Database Tools ............................................................................................................ 46
About This Document

The following annotations have been used to provide additional information.

徕 NOTE
Note provides additional information about the topic.

 tik EXAMPLE
Examples are given throughout the manual to help the reader understand the terminology.

揖 IMPORTANT
This symbol defines items that have significant meaning to the user.

揖* WARNING
The user should pay particular attention to this symbol. It means there is a chance that physical harm could happen to either the person or the equipment.

The following paragraph heading formatting is used in this manual:

1 Heading 1
1.1 Heading 2
1.1.1 Heading 3

This document also uses different body text fonts (listed in Table 0-1) to help you distinguish between names of files, commands to be typed, and output coming from the computer.

<table>
<thead>
<tr>
<th>Font Type</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Courier New Normal</td>
<td>Sample code and screen output</td>
</tr>
<tr>
<td>Courier New Bold</td>
<td>Commands to be typed by the user</td>
</tr>
<tr>
<td>Times New Roman Italic</td>
<td>TinyOS files names, directory names</td>
</tr>
<tr>
<td>Franklin Medium Condensed</td>
<td>Text labels in GUIs</td>
</tr>
</tbody>
</table>
1 Introduction

MOTE-VIEW is designed to be an interface (“client layer”) between a user and a deployed network of wireless sensors. MOTE-VIEW provides users the tools to simplify deployment and monitoring. It also makes it easy to connect to a database, to analyze, and to graph sensor readings.

Figure 1-1 depicts a three-part framework for deploying a sensor network system. The first part is the Mote layer or sensor mesh network. The Motes are programmed with TinyOS firmware (“application”) to do a specific task: e.g., microclimate monitoring, asset tracking, intrusion detection, etc. The second layer or Server layer provides data logging and database services. At this layer sensor readings arrive at the base station (e.g., MIB510, MIB600, or Stargate) and are stored on a server or Stargate. Finally, the third part is the client layer in which software tools provide visualization, monitoring, and analysis tools to display and interpret sensor data. The purpose of this document is to explain the features of MOTE-VIEW and to provide information on the supported Mote layer applications, Mote platforms, and sensor boards.

Figure 1-1. Software framework for a wireless sensor network: The left column represents the wireless sensor network itself. The server layer aggregates the data and allows for a connection to another network or terminal. The client layer is software for viewing and manipulating sensor network data. MOTE-VIEW is a free software tool and is available at www.xbow.com.

1.1 Supported Sensor Boards and Mote Platforms

All of Crossbow’s sensor and data acquisition boards are also supported by MOTE-VIEW (see Table 1-1). MOTE-VIEW supports the MICA-series platforms of wireless sensor network hardware, including the MICA2, MICA2DOT, and MICAz Motes (see Table 1-2). In addition, sensor integrated platforms such as the security/intrusion detection system based on the MSP
Motes and the environmental monitoring system (based on the MEP Motes) can be deployed and monitored (see Table 1-3).

### Table 1-1. Sensor (MTS series) and data acquisition boards supported by MOTE-VIEW and their plug-and-play compatible Mote platforms.

<table>
<thead>
<tr>
<th>Sensor and Data Acquisition Boards</th>
<th>Mote Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MICAz</td>
</tr>
<tr>
<td>MTS101</td>
<td>✓</td>
</tr>
<tr>
<td>MTS300/310</td>
<td>✓</td>
</tr>
<tr>
<td>MTS400/MTS420</td>
<td>✓</td>
</tr>
<tr>
<td>MTS410</td>
<td>✓</td>
</tr>
<tr>
<td>MTS510</td>
<td>✓</td>
</tr>
<tr>
<td>MICA2</td>
<td></td>
</tr>
<tr>
<td>MDA100</td>
<td>✓</td>
</tr>
<tr>
<td>XBW-DA100</td>
<td>✓</td>
</tr>
<tr>
<td>MDA300</td>
<td>✓</td>
</tr>
<tr>
<td>MDA320</td>
<td>✓</td>
</tr>
<tr>
<td>XBW-DA325</td>
<td>✓</td>
</tr>
<tr>
<td>MDA500</td>
<td></td>
</tr>
</tbody>
</table>

### Table 1-2. Mote processor/radio (MPR) platforms supported by MOTE-VIEW.

<table>
<thead>
<tr>
<th>Mote Platforms</th>
<th>Model Number(s)</th>
<th>RF Frequency Band(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICAz</td>
<td>MPR2400</td>
<td>2400 MHz to 2483.5 MHz</td>
</tr>
<tr>
<td>MICA2</td>
<td>MPR400</td>
<td>868 MHz to 870 MHz; 903 MHz to 928 MHz</td>
</tr>
<tr>
<td></td>
<td>MPR410</td>
<td>433.05 to 434.8 MHz</td>
</tr>
<tr>
<td></td>
<td>MPR420</td>
<td>315 MHz (for Japan only)</td>
</tr>
<tr>
<td>MICA2DOT</td>
<td>MPR510</td>
<td>868 MHz to 870 MHz; 903 MHz to 928 MHz</td>
</tr>
<tr>
<td></td>
<td>MPR510</td>
<td>433.05 to 434.8 MHz</td>
</tr>
<tr>
<td></td>
<td>MPR520</td>
<td>315 MHz (for Japan only)</td>
</tr>
</tbody>
</table>

### Table 1-3. Sensor integrated (MEP, MSP) platforms supported by MOTE-VIEW.

<table>
<thead>
<tr>
<th>Sensor Integrated Mote Platforms</th>
<th>Description of Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEP410</td>
<td>Microclimate and ambient light monitoring</td>
</tr>
<tr>
<td>MEP510</td>
<td>Temperature and humidity monitoring</td>
</tr>
<tr>
<td>MSP410</td>
<td>Physical security and intrusion detection</td>
</tr>
</tbody>
</table>

### 1.2 Supported Mote Software Applications

*MOTE-VIEW* can support a number of different firmware applications running on the Motes that make up the mesh network:
NOTE: All the Mote applications mentioned below can be programmed through the MoteConfig tool described in Section 5.2.6, if the application is precompiled. Only Surge_Reliable, Surge_Reliable_Dot, and the XMesh-enabled applications come precompiled with MOTE-VIEW. Each Mote must be programmed with a unique identification number (“node ID”) when deploying a network of greater than one node.

1. Surge_Reliable and Surge_Reliable_Dot: Surge_Reliable and Surge_Reliable_Dot are the standard applications that are preprogrammed into Motes shipped in a Crossbow MOTE-KIT. If MOTE-VIEW was installed into the default directly, then pre-compiled code can be found in \C:\Program Files\Crossbow\MoteView\xmesh\<platform>\, where <platform> = micaz, mica2, or mica2dot.

For the MICA2 and MICA2DOT there are two versions of Surge_Reliable: One with a networking stack that doesn’t use power managed (“high-power” or “hp”) networking and another with power managed and time synchronized (“low-power” or “lp”) networking. For the MICAz, currently there is only a high powered networking version.

Table 1-4. Pre-compiled Surge_Reliable applications for the MICAz (MPR2400), MICA2 (MPR4x0), and MICA2DOT (MPR5x0) available with the MOTE-VIEW install.

<table>
<thead>
<tr>
<th>Surge Multihop Applications</th>
<th>Platform</th>
<th>MPR Model</th>
<th>Binary file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICAz</td>
<td>MPR2400</td>
<td>Surge_2420_hp.exe</td>
<td></td>
</tr>
<tr>
<td>MICA2</td>
<td>MPR400</td>
<td>410</td>
<td>420</td>
</tr>
<tr>
<td>MICA2DOT</td>
<td>MPR500</td>
<td>510</td>
<td>520</td>
</tr>
</tbody>
</table>

xxx = 315, 433, or 915. <mode> = hp or lp. hp = high-power mesh networking. lp = low-power mesh networking via low-power listening and time synchronized data transmissions.

These applications are compatible with the Surge-View analysis tools described in Crossbow’s Getting Started Guide, which has the instructions on how to compile and download these applications using the Cygwin interface.

2. XMesh enabled applications: XMesh is Crossbow’s multihop mesh networking protocol that has various options including low-power listening, time synchronization, sleep modes, any-to-base and base-to-any routing. All of our sensor and data acquisition boards are supported with XMesh enabled applications. The tables below are a summary of the XMesh applications for the corresponding sensor boards.

Table 1-5. Pre-compiled MICAz XMesh applications available in the MOTE-VIEW install.

<table>
<thead>
<tr>
<th>MICAz Mote (MPR2420)</th>
<th>Binary file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board Model</td>
<td></td>
</tr>
<tr>
<td>MTS boards</td>
<td></td>
</tr>
<tr>
<td>MTS101</td>
<td>XMTS101_2400_hp.exe</td>
</tr>
<tr>
<td>MTS300</td>
<td>XMTS300_2400_hp.exe</td>
</tr>
<tr>
<td>MTS310</td>
<td>XMTS310_2400_hp.exe</td>
</tr>
<tr>
<td>MTS400</td>
<td>XMTS400_2400_hp.exe</td>
</tr>
<tr>
<td>MTS410</td>
<td>XMTS410_2400_hp.exe</td>
</tr>
<tr>
<td>MTS420</td>
<td>XMTS420_2420_hp.exe</td>
</tr>
<tr>
<td>MDA board</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>MDA100</td>
<td>XMDA100_2400_hp.exe</td>
</tr>
<tr>
<td>XBW-DA100</td>
<td>XBW-DA100_2400_hp.exe</td>
</tr>
<tr>
<td>MDA300</td>
<td>XMDA300_2400_hp.exe</td>
</tr>
<tr>
<td>MDA320</td>
<td>XMDA320_2400_hp.exe</td>
</tr>
<tr>
<td>XBW-DA325</td>
<td>XBW-DA325_2400_hp.exe</td>
</tr>
<tr>
<td>Base Station (common for all sensorboards)</td>
<td></td>
</tr>
<tr>
<td>XMeshBase</td>
<td>XMeshBase_2420_hp.exe</td>
</tr>
</tbody>
</table>

Table 1-6. Pre-compiled MICA2 XMesh Applications Available in MOTE-VIEW

<table>
<thead>
<tr>
<th>MICA2 Mote (MPR4x0, x = 0, 1, or 2)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Board Model</td>
<td>Binary file name</td>
</tr>
<tr>
<td>MTS boards</td>
<td></td>
</tr>
<tr>
<td>MTS101</td>
<td>XMTS101_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MTS300</td>
<td>XMTS300_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MTS310</td>
<td>XMTS310_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MTS400</td>
<td>XMTS400_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MTS410</td>
<td>XMTS410_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MTS420</td>
<td>XMTS420_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MDA board</td>
<td></td>
</tr>
<tr>
<td>MDA100</td>
<td>XMDA100_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MDA300</td>
<td>XMDA300_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MDA320</td>
<td>XMDA320_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>Base Station (common for all sensorboards)</td>
<td></td>
</tr>
<tr>
<td>XMeshBase</td>
<td>XMeshBase_xxx_&lt;mode&gt;.exe</td>
</tr>
</tbody>
</table>

xxx = 315, 433, or 915. <mode> = hp or lp. hp = high-power mesh networking. lp = low-power mesh networking via low-power listening and time synchronized data transmissions.

<table>
<thead>
<tr>
<th>MICA2DOT Mote (MPR5x0, x = 0, 1, or 2)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Board Model</td>
<td>Binary file name</td>
</tr>
<tr>
<td>MTS boards</td>
<td></td>
</tr>
<tr>
<td>MTS510</td>
<td>XMTS510_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>MDA boards</td>
<td></td>
</tr>
<tr>
<td>MDA500</td>
<td>XMDA500_xxx_&lt;mode&gt;.exe</td>
</tr>
<tr>
<td>Base Station (common for all sensorboards)</td>
<td></td>
</tr>
<tr>
<td>XMeshBase_Dot</td>
<td>XMeshBase_Dot_xxx_&lt;mode&gt;.exe</td>
</tr>
</tbody>
</table>

xxx = 315, 433, or 915. <mode> = hp or lp. hp = high-power mesh networking. lp = low-power mesh networking via low-power listening and time synchronized data transmissions.
3. \textit{XSensor} applications: These are test applications for Crossbow’s sensor and data acquisition boards and allows the user to quickly and easily test sensor and data acquisition boards when attached to Mote. These applications send the output over the Mote’s UART thereby allowing the user to test these boards without using an RF link. Testing by RF link can also be done by programming another Mote with \textit{TOSBase} and attaching it to an MIB510, MIB600, or Stargate gateway.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Sensor and Data Acquisition Boards & Application Name & Location of Driver Folder \\
\hline
MTS101 & XSensorMTS101 & tinyos-1.x/tos/sensorboards/basicsb/ \\
MTS300/310 & XSensorMTS300 & tinyos-1.x/contrib/xbow/tos/sensorboards/mts310/ \\
MTS400/MTS420 & XSensorMTS400 & tinyos-1.x/contrib/xbow/tos/sensorboards/mts400/ \\
MTS510 & XSensorMTS510 & tinyos-1.x/contrib/xbow/tos/sensorboards/mts510/ \\
MDA300 & XSensorMDA300 & tinyos-1.x/contrib/xbow/tos/sensorboards/nda300/ \\
MDA500 & XSensorMDA500 & tinyos-1.x/tos/sensorboards/basicsb/ \\
\hline
\end{tabular}
\caption{Sensor and data acquisition boards and the corresponding XSensor application}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
MSP410 Mote Security Package & \\
\hline
\textit{MSP410\_433\_base.exe} & To be used on an MPR410 MICA2 Mote assigned with node ID of 0 (“base node”). \\
\hline
\textit{MSP410\_433\_hp.exe} & For the MSP410 Mote \\
\hline
\end{tabular}
\caption{Pre-compiled MSP XMesh applications available in MOTE-VIEW}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
MEP410 Mote Environmental Package & \\
\hline
\textit{MEP410\_433\_<mode>.exe} & For the MEP410 and an MPR410 MICA2 Mote when assigned a node ID of 0 (“base node”). \\
\hline
\textit{MEP510\_433\_<mode>.exe} & For the MEP510 Mote. \\
\hline
\end{tabular}
\caption{Pre-compiled MEP XMesh applications available in MOTE-VIEW}
\end{table}

\textbf{NOTE:} These \textit{XSensor} applications are not pre-compiled in \textit{MOTE-VIEW}. Instructions on how to compile these applications are in Crossbow’s \textit{Getting Started Guide}. 
2 Installation

2.1 Supported PC Platforms and Operating Systems

MOTE-VIEW is supported on the following platforms:

- Windows XP Home
- Window XP Professional
- Windows 2000 with SP4

The screen resolution must be at least $800 \times 600$ or the interface would require scrollbars.

2.1.1 PC Interface Port Requirements

There are three different interface port requirements depending on the gateway platform used as the base station in a sensor network.

1. For an MIB510 serial gateway: an RS-232 serial port or a USB port and a USB-to-serial converter that is compatible with the PC and its operating system.
2. For an MIB600 Ethernet gateway: A wired Ethernet or 802.11 wireless card only if the MIB600 is on a LAN with wireless access.
3. For a Stargate gateway or other server: A wired Ethernet, an 802.11 wireless card only if the Stargate has a wireless modem or is on a LAN with wireless access, or a cellular modem for wireless Internet access.

2.2 Additional Software Requirements

For the application to run the following additional components are required:

- PostgreSQL 8.0 database service
- PostgreSQL ODBC driver
- Microsoft .NET framework

The installation files for these components are included on the MOTE-VIEW installation CD.

2.2.1 PostgreSQL

All the visualization tools in MOTE-VIEW require being connected to a database. This database can reside on your PC (“localhost”), a remote server, or a Stargate. The size of this database is bound by the storage available on the system.

The installation of PostgreSQL will automatically install and configure a local PostgreSQL 8.0 database on the machine when that option is checked. The installation requires administrative privileges on the system, including the ability to create a new user called postgres (for PostgreSQL).
**IMPORTANT**: If your PC is to host both the server and client layer functions, then running the *PostgreSQL* database service is required to use *MOTE-VIEW*. However, if you are accessing a server or Stargate that is running *XServe/PostgreSQL*, then you don’t need to run the service on your PC.

### 2.3 Installation Steps

Shut down all the programs running on your computer.

1. Insert the *Wireless Sensory Systems Support* CD into the computer’s CD drive.
2. Double-click on *MoteViewSetup.exe* from *MOTE-VIEW* folder.
3. At the **Welcome to the MOTE-VIEW Setup Wizard** window, click **Next>**.
4. Select the desired installation directory, then click **Next>** (see Figure 2-1).

![Step 4: Destination folder select and confirm screen.](image)

*Figure 2-1. Step 4: Destination folder select and confirm screen.*
5. Select the desired Start Menu folder name and click **Next>**.

![Screenshot](image1)

*Figure 2-2. Step 5: Screenshot for selecting the Start Menu folder for MOTE-VIEW.*

6. Select all available installation tasks and click **Next>** (see Figure 2-3).

![Screenshot](image2)

*Figure 2-3. Step 6: Select all options if installing MOTE-VIEW for the first time.*
7. Confirm your selections and click **Install**.

![Step 7: Confirm your selections with this screen.](image)

Figure 2-4. Step 7: Confirm your selections with this screen.

Execute the .NET framework and *PostgreSQL* OBDC installation prompts if applicable and follow the instructions.

⚠ **NOTE:** If you have a version of MOTE-VIEW prior to v1.0 you may get a *PostgreSQL 8.0.0-rc1* window (Figure 2-5) when installing the database. Record the exact error message in `C:\Program Files\PostgreSQL\8.0.0-rc1\tmp\initdb.log`. In most cases you can simply click the **OK** button and proceed. However, in other cases you may have to do the following:

a. Uninstall *PostgreSQL 8.0* from **Start**>**Control Panel**>**Add or Remove Programs**

b. Remove the `C:\Program Files\PostgreSQL\` directory manually through the Windows Explorer utility (**Start**>**Right Click**>**Explore**).

c. Resinstall *MOTE-VIEW* with all checked **except** the .NET Framework.

![Error window that may occur during the PostgreSQL Installation.](image)

Figure 2-5. Error window that may occur during the PostgreSQL Installation.

8. Please note that if you receive an MDAC warning such as the one pictured below, you may ignore this warning and continue on with the installation. The warning is simply notification that your operating system contains an MDAC version that is newer than what *MOTE-VIEW* is installing.
9. When the Setup Wizard has finished it will ask if you want to start MOTE-VIEW. You may start MOTE-VIEW now, but in some cases it may ask you to restart your computer first.

10. If you have restarted or at any time after you can start MOTE-VIEW by double-clicking on the MOTE-VIEW icon or by going to the installation folder (the default install folder is C:\Program Files\Crossbow\MoteView\) and clicking on MoteView.exe.

11. (Optional) Connect the base station of your choice
   - MIB510 with Mote to the PC’s serial port. Alternatively, use the USB port if you have a compatible USB-to-serial converter and driver.
   - MIB600 with a Mote to a LAN or LAN hub or to a PC directly by using a crossover Ethernet cable.
   - Stargate with a Mote to a LAN, to the Internet, or to a PC directly by using a crossover Ethernet cable.
3 Application Quick Start

Once a sensor network is running and MOTE-VIEW is installed on a PC, there is minimal configuration necessary to start collecting data from the sensor network.

⚠️ IMPORTANT: There are two ways to repair a corrupt database installation:

1) Run the installer again and select only “Reset Mote Database”.
2) Open a File Manager, navigate to C:\Program Files\Crossbow\MoteView, and execute “resetdb.cmd”.

3.1 Connecting to the Demo Database on Your PC

During the installation of MOTE-VIEW a static database was included to make it possible to demonstrate MOTE-VIEW’s features without having to be connected to an active sensor network or to be connected to a remote server/database. The steps described here also apply to viewing data collected during a connection to an active sensor network.

1. Click on the green connect icon, in the upper left hand corner or click on File > Connect to Remote Database/Stargate. The Database Server Configuration will appear (see Figure 3-1).

2. Check that the pull down item next to Database = task and that Table Name = sample_mts310.
3. Click on Apply. MOTE-VIEW will then access the table in the database task. Afterwards the screen will display the data from the database table file sample_mts310.
Figure 3-2. After MOTE-VIEW connects to a database or after starting a data log, data starts to appear in the MOTE-VIEW Data view. This screenshot is from the static database, sample_mts310.

3.2 Connecting to an Active Sensor Network Using an MIB510 or MIB600

The first step typically is to connect to either a gateway device—MIB510, MIB600, or Stargate—which will send data to your PC (“localhost server”) or a server or Stargate that’s running a Postgres database.

1. On the menu bar click on File > Connect to MIB510/MIB600.

2. If using a MIB510, make sure that the Serial Port: COM is set to the correct port number and that baud is 57600.

3. If using a MIB600, select the button next to Host and in the box next to it enter the IP address of the MIB600. The Port should default to mib600.
EXAMPLE 3-1:

Screenshots of the Connect to MIB510/MIB600 popup window. On the left is an example of connecting to an MIB510 on COM1. The baud rate is set to mica2 which is equivalent to a data rate of 57600 bps. On the right is an example of connecting to an MIB600 assigned with IP address 10.1.1.219, which is connected to the LAN.

4. Choose the XMesh application that matches to that which the Mote has been programmed.

<table>
<thead>
<tr>
<th>If you have an XMesh Application...</th>
<th>...or if you have an XSensor application</th>
<th>Then choose one of the following pull down choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default. Use when you don’t see the XMesh application</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Surge_Reliable, Surge_Reliable_Dot</td>
<td>XSensorMTS101</td>
<td>Surge</td>
</tr>
<tr>
<td>XMTS101_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorMTS300</td>
<td>MTS101</td>
</tr>
<tr>
<td>XMTS300_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorMTS310</td>
<td>MTS300</td>
</tr>
<tr>
<td>XMTS310_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorMTS400</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS400_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorMTS410</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS410_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorMTS420</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS420_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorMTS510</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS510_&lt;freq&gt;_&lt;mode&gt;</td>
<td>MDA100</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS511_&lt;freq&gt;_&lt;mode&gt;</td>
<td>MDA100</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS512_&lt;freq&gt;_&lt;mode&gt;</td>
<td>MDA100</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS513_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorMDA300</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS514_&lt;freq&gt;_&lt;mode&gt;</td>
<td>MDA300</td>
<td>MTS310</td>
</tr>
<tr>
<td>XMTS515_&lt;freq&gt;_&lt;mode&gt;</td>
<td>MDA300</td>
<td>MTS310</td>
</tr>
<tr>
<td>MBW-DA100_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorDA100</td>
<td>MTS310</td>
</tr>
<tr>
<td>MBW-DA300_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorDA300</td>
<td>MTS310</td>
</tr>
<tr>
<td>MBW-DA320_&lt;freq&gt;_&lt;mode&gt;</td>
<td>MDA300</td>
<td>MTS310</td>
</tr>
<tr>
<td>MBW-DA325_&lt;freq&gt;_&lt;mode&gt;</td>
<td>XSensorDA325</td>
<td>MTS310</td>
</tr>
</tbody>
</table>
**IMPORTANT:** The Log to Database option must be selected when connecting to a MIB510 or MIB600 for viewing live sensor network data.

![Advanced Logging Options Window](image)

**Figure 3-3. Screenshot of the Advanced Logging options window that appears after clicking on the Advanced... button. By default Log to Database and Spawn Separate Shell are selected.**

5. Table 3-1 for a description of the other logging options. Click on OK when finished selecting option.

<table>
<thead>
<tr>
<th>Advanced Logging Option</th>
<th>Description</th>
<th>XServe Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Raw Data</td>
<td>The raw packet bytes coming into the PC.</td>
<td>-r</td>
</tr>
<tr>
<td>Display Parsed Data</td>
<td>The raw ADC values displayed as parsed fields.</td>
<td>-p</td>
</tr>
<tr>
<td>Display Converted Data</td>
<td>The incoming data converted to engineering units.</td>
<td>-c</td>
</tr>
<tr>
<td>Log to Database</td>
<td>Will store raw ADC values to PostgreSQL database.</td>
<td>-l</td>
</tr>
<tr>
<td>Full Version Information</td>
<td>Display exact version for all board modules</td>
<td>-v</td>
</tr>
<tr>
<td>Spawn Separate Shell</td>
<td>Launches XServe in command line shell</td>
<td>n/a</td>
</tr>
</tbody>
</table>

6. If you are not able to receive data, you may also need to select the Live check box on the main MOTE-VIEW screen if it has not been previously selected. Also check to see if data is coming in from your nodes by viewing the Server Messages pane at the bottom of your MOTE-VIEW display or by looking at the XServe shell.

### 3.3 Connecting to a Database (Local or Remote) Server or a Stargate

To connect to the Sensor Network,

1. Click on the green Connect icon, in the toolbar menu. As necessary fill out the information in top four boxes the Database Server Configuration dialog window. The minimal configuration requires information for the Server, Port, User, and Password.

**NOTE:** Users should be advised that the Server, Port, Username, and Password fields are preset to reasonable defaults and should not have to be changed.
2. Click the **Connect** button to connect *MOTE-VIEW* to the given server.

3. Select the **Database** and then **Table Name** from the drop down menu.

4. Click on the **Apply** button to store your selections and begin viewing data from the database. This configuration will be stored in a short pull down list to the right of the **Connect** icon. This quick connect list will remember the last table used for each **Server-Database** pair.

### 3.4 Auto-discovery of Nodes

New nodes will also appear in the upper left hand corner of the **Topology** view. These nodes can be dragged to their correct position on the topology map and those placements can be saved to the configuration file in the database.
4 User Interface Functional Description

4.1 MOTE-VIEW at a Glance

MOTE-VIEW has four main user interface sections:

1. Toolbar / Menus: Allows the user to specify actions and initiate command dialogs.
2. Node List: Shows all known nodes in a deployment and health status summary.
3. Visualization Tabs: Enables the user to view the sensor data in various ways.
4. Server Messages: Displays a log of server events and incoming messages.

4.2 Node List

The Node List displays all the known nodes in a deployment.

Table 4-1. Features and icon properties of the node list.

<table>
<thead>
<tr>
<th>Node List Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkbox next to node id</td>
<td>Select nodes to be plotted in the Charts view</td>
</tr>
<tr>
<td>Change node properties</td>
<td>Double click on a node in the list or right click and choose Properties.</td>
</tr>
<tr>
<td>Sort data</td>
<td>Click on the column header. Static display only when not in &quot;live&quot; mode.</td>
</tr>
<tr>
<td>Add nodes</td>
<td>Right click on any node or use the auto-detect feature as results arrive.</td>
</tr>
</tbody>
</table>

Gray Mote icon: No results received
Green Mote icon: Fresh results within the last 20 minutes*
Moss (light green) Mote icon: Results stale by >20 minutes*
Yellow Mote icon: Results stale by >40 minutes*
Orange Mote icon: Results stale by >60 minutes*
Red Mote icon: Results stale by more than a day*

*The time intervals for the node color changes cannot be edited in MOTE-VIEW 1.2.

4.3 Visualization Tabs

The visualization tabs provide three ways to view your sensor data. The main display of the user interface consists of the four tabs: 1) Data, 2) Command, 3) Charts, and 4) Topology.

4.3.1 Data View

The Data tab displays the latest sensor readings received for each node in the network (see Figure 4-1 for an example from the demo database). Any column of data can be sorted by left clicking the top. This allows you to sort by node ID, parent, temperature, voltage, last result time, or any other sensor reading.

The columns include: Id, <sensed value 1>, <sensed value 2>, ..., <sensed value N>, and Time. The sensor data is automatically converted into standard engineering units.
4.3.2 Chart View

The Chart view provides the ability to generate graphs of a sensor reading against time for some set of nodes. The following features and constraints apply to the graphs plotted in this view.

- Up to 3 sensor types can be selected for plotting, i.e., 3 different graphs.
- Up to 24 different Nodes can be selected for plotting. A different color code is to be used for each node, with a legend on the right side of the window.
- The $y$-axis on the graph shows the time.
- The $x$-axis on the graph shows data in engineering units for the sensor readings.
Figure 4-2. Screenshot of the demo database in Chart view.

- The user can zoom into and pan through data, as instructed in Table 4-2, within each of the graphs independently of each other.

Table 4-2. How to zoom, pan, and reset in the Chart View.

<table>
<thead>
<tr>
<th>Desired Action</th>
<th>Instructions</th>
</tr>
</thead>
</table>
| To zoom in           | Hold down the “Shift” key.         
                       | Left click and drag a region to zoom into.      
                       | Release the mouse to complete the region selection.  
                       | Release the “Shift” key. |
| To pan through data | Hold the “Ctrl” key               
                       | Left click and drag a point within the chart to the new location.       
                       | Release the mouse button. |
| To zoom out fully    | Go to the Tools menu and select “Refresh View.”                         |
| To undo a zoom       | Hold down the “Shift” key        
                       | Right click once for each level of undo                      |
| To undo a pan        | Hold down the “Ctrl” key         
                       | Right click once for each level of undo                      |
4.3.3 Topology View

The Topology view shows a map of the network of Motes, including placement and parenting information. This allows the user to define and view a topology of their Mote deployment.

- New nodes will show up in the upper left hand corner.
- The user can drag nodes and place them at a new location on the map with a left click of the mouse. Node locations are stored in the database and are shared by all users of that database.

Figure 4-3. Screenshot of the demo database in Topology view. The temperature values have been selected. The topology view defaults to false coloring and can be turned off. The legend in the lower right appears only when the MOTE-VIEW is maximized.
A visualization pop-up menu will appear when the user right clicks the background bitmap.

**Table 4-3. Items from the visualization pop-up menu that appears when right-clicking in the map area of the Topology view.**

<table>
<thead>
<tr>
<th>Visualization Menu</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Node</td>
<td>Creates a new node at the current mouse location. A Mote properties dialog will appear to allow selection of node name and id number.</td>
</tr>
<tr>
<td>Arrange Nodes &gt; Grid</td>
<td>Will automatically arrange all the nodes into a grid layout. This is useful when there are a large number of unplaced nodes.</td>
</tr>
<tr>
<td>Save Node Properties</td>
<td>Saves the current node placement and naming into the database.</td>
</tr>
<tr>
<td>Revert Node Properties</td>
<td>Reloads node placement and naming from the database.</td>
</tr>
<tr>
<td>Visualization Properties</td>
<td>Allows the user to visualize temperature gradients or other Mote properties using a specific color pattern.</td>
</tr>
<tr>
<td>MSP Properties</td>
<td>Sets the orientation of quadrant 1 relative to image in the topology view and the draw radius of a PIR event</td>
</tr>
<tr>
<td>Load Bitmap</td>
<td>Allows users to select a bitmap from their file system to be the background of the map. Supported background image formats include: .bmp, .gif, .ico, and .jpg. The image will be automatically scaled to fit the available screen space.</td>
</tr>
<tr>
<td>Use Default Bitmap</td>
<td>Use the standard grid as the background of the topology view.</td>
</tr>
</tbody>
</table>

The Visualization Properties lets users customize the following:

**Sensor Color Gradient** – Allows users to specify the minimum and maximum sensor values and associate specific color with it.

- Right-click on the Topology bitmap and select Visualization Properties.
- Click on the Sensor tab of the Visualization Properties
- Select a Sensor type from the drop-down list
- Specify desired Maximum and Minimum sensor values
- Click on the square colored box next to Maximum Color or Minimum Color
- This will bring up another dialog box
- Move the square marker on to choose the desired color and click OK.

The Sensor info also lets users display the sensor readings next to the node on the topology map and specify the font for the text.
**Mote Visualization** – Users can choose from 3 different display styles for nodes.

- **None** – No nodes are displayed
- **BlackDot** – A black dot is displayed to represent the node
- **MoteGlow** – A colored circle is displayed along with the node ID.

Users can also check options for Draw Links between nodes and specifying whether Gateway has a sensor on it for gradient visualization.
**Isobar Visualization** – Users can specify Isobar style

- **None** – No Isobar visualization
- **Draw Gradient** – Draws the color gradient based on the sensor readings;

Users can also specify the Mote Radius (specified as percentage of bitmap image size; 0=0%, 10=100%) for spread of gradient around the node; check options for displaying the ScaleBar Legend and drawing the Logo.
**Health Visualization** - Users can specify the time duration after which the link between the nodes goes grey. If a packet is not received from any after the specified minutes, the link would turn grey.
MOTE-VIEW displays a colored circle around each node to indicate temperature values. The default values are listed in Table 4-4.

![Figure 4-7. Screenshot for Link Quality Visualization Dialog](image)

**Table 4-4. Color and Temperature Range Mapping**

<table>
<thead>
<tr>
<th>Color</th>
<th>Temperature Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red:</td>
<td>&gt; 30 °C</td>
</tr>
<tr>
<td>Orange:</td>
<td>26 °C to 30 °C</td>
</tr>
<tr>
<td>Yellow:</td>
<td>22 °C to 26 °C</td>
</tr>
<tr>
<td>Green:</td>
<td>18 °C to 22 °C</td>
</tr>
<tr>
<td>Blue:</td>
<td>&lt; 18 °C</td>
</tr>
</tbody>
</table>

The user can bring up a menu of actions and bring up the Mote properties dialog with a right click of the mouse. By rolling over a node with the mouse, the user can bring up a status box with recent sensor readings of the Mote.
4.3.4 Live Mode

In order to have MOTE-VIEW display data from an active sensor network, the users must check on the “Live” checkbox. In Live mode, MOTE-VIEW refreshes the node list, charts, and topology views at a regular interval defined in the Tools>Settings... dialog. The default interval is 10 seconds.

**NOTE:** The optimal setting for **Update Interval** is one half the sample rate used by the nodes.

4.3.5 Command

The **Command** tab provides the user with an ability to change different node parameters wirelessly.

**NOTE:** These commands do not get stored in the EEPROM and hence are volatile.

To change the System configuration do the following:

1. Click on **System** tab on the left side of the Command Window.
2. Select the Node ID and Group ID of your network. If changing the system parameters for all nodes, check the “All Nodes” checkbox.
3. From New Configuration, select either
   - **Data Rate** – Sets the Update rate between 300 msec and 2147483647 msec
   - **Node ID** – Changes the node ID between 1 and 65534
   - **Radio Power** – Sets the RF power between 0 (min) and 255 (max)
4. Click on Set to send the specified commands to node(s) in your network.
To change the LED status, do the following.

1. Click on **System** tab on the left side of the Command Window.
2. Select the Node ID and Group ID of your network. If changing the system parameters for all nodes, check the “All Nodes” checkbox.
3. Select the color of LED(s) you are interested (Red, Yellow or Green) and check:
   - **ON** – To turns the LED On.
   - **OFF** – To turns the LED Off
   - **TOGGLE** – To toggle the state of the LED
4. Click on Set to send the specified commands to node(s) in your network.
5. You can click SET ALL to activate all LEDs simultaneously.
Figure 4-9. Screenshot of the LED actuation in Command Tab.

**NOTE:** The Command feature is currently supported only for high power (hp) mesh networking applications.
5 Menus

There are two sections in the MOTE-VIEW menu area. The top part is the horizontal arrangement of the five menu items: 1) **File**, 2) **Tools**, 3) **Units**, 4) **Window**, and 5) **Help**. Below this is a horizontally arranged icon toolbar, which provides quick access to some of the more common features of the MOTE-VIEW application. From left to right, the toolbar provides buttons for database connection, data logging with XServe, MoteConfig, Alerts Manager, MOTE-VIEW settings, export to CSV, print, and print preview, refreshing the view, and displaying live (vs. historical) sensor network data. This chapter will describe the features in each of these menu items.

![Figure 5-1. Screenshot showing a portion of the MOTE-VIEW window to highlight the icon toolbar menu.](image)

5.1 FileConnect

This will bring up the Server Configuration dialog.

5.1.1 Export

You can export data in two different formats: XML or CSV (comma delimited text)

5.1.2 Print Preview

The Print Preview screen will display the expected output for the Print command. The current visualization tab is the one that is printed: Data, Chart, or Topology view. The magnifying glass icon allows you to zoom in on the preview. The printer icon will immediately print out the page to the default printer.

5.1.3 Print

The Print dialog is a standard Windows dialog that allows you to select the printer to which print output will be sent. The current visualization tab is the one that is being printed: Data, Chart, or Topology.

5.1.4 Exit

This menu option will exit the application and close MOTE-VIEW. MOTE-VIEW always attempts to save the current configuration when closed and restore that configuration the next time the application is started.
5.2 Tools

5.2.1 Settings

The Settings menu will bring up a dialog that allows the user to set MOTE-VIEW global preferences.

![Figure 5-2. The Settings dialog to change the Update Interval (sec) and the Chart Resolution (%).](image)

Currently this dialog provides access to two settings in MOTE-VIEW:

- **Update Interval (sec)**: Specifies the length of time in seconds between database updates in “Live” mode.

  **NOTE**: The optimal Update Interval is one half of the rate which the nodes are transmitting sensor data. This does not affect the rate at which the Motes are sending data into the network.

- **Chart Resolution (%)**: Specifies what percentage of data to draw from the database when creating charts. The lower the percentage, the faster the refresh and graphical update rates. That is, the trade off is between data resolution and increased graphics performance.

5.2.2 Add Node

This will bring up the Mote Properties dialog for creation of a new node. The user can select a unique node ID and name the node at this time.
5.2.3 Refresh View
This will force a visual refresh of all visualization windows in MOTE-VIEW. After clicking on this menu the Nodes list, the Chart, and Topology views will all update to display the latest information from the sensor network.

5.2.4 Log Data
The data must be actively logging into a database before live data can be displayed in MOTE-VIEW. The File > Connect to MIB510/MIB600 dialog (see Figure 5-4) provides a means to start and stop logging of Mote data on the user’s PC (“localhost”) using the XServe software tool.
Figure 5-4. The Connection to MIB510/MIB600 dialog. The screenshot on the left is an example for MIB510 users and the screenshot on the right is an example for MIB600 users. MIB510 users need to know the Serial Port: COM number to which the MIB510 is attached. For Baud the 57600 or mica2 is typical. MIB600 users need to input the IP address of the MIB600.

5.2.5 Advanced Logging

Table 5-1. Description of advanced logging options and the equivalent XServe flag.

<table>
<thead>
<tr>
<th>Advanced Logging Option</th>
<th>Description</th>
<th>XServe flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Raw Data</td>
<td>The raw packet bytes coming into the PC.</td>
<td>-r</td>
</tr>
<tr>
<td>Display Parsed Data</td>
<td>The raw ADC values displayed as parsed fields.</td>
<td>-p</td>
</tr>
<tr>
<td>Display Converted Data</td>
<td>The incoming data converted to engineering units.</td>
<td>-c</td>
</tr>
<tr>
<td>Log to Database</td>
<td>Will store raw ADC values to PostgreSQL database.</td>
<td>-l</td>
</tr>
<tr>
<td>Full Version Information</td>
<td>Display exact version for all board modules</td>
<td>-v</td>
</tr>
<tr>
<td>Spawn Separate Shell</td>
<td>Launches XServe in command line shell</td>
<td>n/a</td>
</tr>
</tbody>
</table>

By selecting the Log to Database check box and clicking on the OK button you will be able to receive live data from your nodes. Please note that you may also need to select the Live checkbox on the main MOTE-VIEW screen if it has not been previously selected.
If you are not able to receive data, you should check to see if your nodes are transmitting by viewing the data logging screen at the bottom of your MOTE-VIEW display. Depending upon the selections you made in the Data Logging window, you may see data in raw, converted or parsed format.

![Portion of a Command Prompt window that appears once the data logging has started.](image)

The Start button will be enabled if XServe is not already running. Clicking on Start will start XServe running as a separate process with the given options. The output of XServe will be streamed to the Server Messages pane unless Spawn Separate Shell has been selected in which case a Command Prompt window, as shown above in Figure 5-5, will be created. Sample outputs from various Crossbow applications are listed below.

**XMTS101 Logging Output**

```
INSERT into mts101_results
(result_time,nodeid,parent,voltage,temp,light) values
(now(),0,126,377,183,186)
```

**XMTS310 Logging Output**

```
INSERT into mts310_results
(result_time,nodeid,parent,voltage,temp,light,accel_x
,accel_y,mag_x,mag_y,mic) values
(now(),0,126,378,197,301,193,177,210,215)
```

**XMTS400 / XMTS420  Logging Output**

```
INSERT into mts420_results
(result_time,nodeid,parent,voltage,humid,humtemp,intermacal1
,intersemacal2,intersemacal3,intersemacal4,prtemp,press)
values (now(),
0,126,377,1391,6517,45446,46808,38941,45989,26835,18130)
```
### XMEP410 Logging Output

```sql
INSERT into enviro_results
(result_time, nodeid, parent, epoch, voltage, humid, humtemp, inthum, inttemp, accel_x, accel_y, photo1, photo2, photo3, photo4, prtemp, press) values
(now(), 0, 126, 0, 379, 0, 0, 65535, 65535, 201, 0, 174, 200, 207, 0, 65535, 65535)
```

### XMTS510 Logging Output

```sql
INSERT into mts510_results
(result_time, nodeid, parent, light, accel_x, accel_y, mic) values
(now(), 0, 126, 273, 222, 205, 245)
```

### XMDA300 Logging Output

```sql
INSERT into mda300_results
(result_time, nodeid, parent, voltage, humid, humtemp, adc0, adc1, adc2, digi0, digi1, digi2) values
(now(), 0, 126, 377, 960, 6678, 2500, 2243, 2505, 0, 0, 0)
```

### XMDA500 Logging Output

```sql
INSERT into mda500_results
(result_time, nodeid, parent, voltage, temp, adc2, adc3, adc4, adc5, adc6, adc7) values
(now(), 0, 126, 191, 527, 306, 240, 213, 191, 234)
```

### XMEP510 Logging Output

```sql
INSERT into enviro_results
(result_time, nodeid, parent, epoch, voltage, therm, humid, humtemp) values
(now(), 0, 126, 15, 376, 517, 65535, 65535)
```

### 5.2.6 Programming Motes with MoteConfig

*MoteConfig* is a dialog utility for programming Motes by providing an interface for downloading pre-compiled TinyOS software applications. A major benefit of *MoteConfig* is that Mote platforms can be programmed without having to install the TinyOS programming environment. Click on **Settings > Interface Board Settings**... and choose the proper MIB gateway and check that the port settings are correct.
EXAMPLE 5-1: Left screenshot shows Interface Board Settings for an MIB510 on COM port 1. Right screenshot shows Interface Board Settings for an MIB600 with IP address 10.1.1.248 and on the same LAN as that of the PC.

Select your desired Node ID, Group ID, Radio Band, Channel Number, and RF Power from the drop down menus.

13. Right click on the Select... button on the top right which will open up a directory window (see Figure 5-6). Or navigate to another directory which has the pre-compiled Mote application you wish to install. Select the specific .exe file of the application you want to download into the Mote.

![Figure 5-6. Screenshot for selecting XMesh applications.](image)

NOTE: The XMesh applications distributed with MOTE-VIEW 1.2 now require XMeshBase_xxx.exe programmed with MOTE ID=0 as base station application.

14. Right Click on the Program button and you will see the resulting output at the bottom of the screen (see Figure 5-7).
Figure 5-7. Screenshot of MoteConfig which has successfully installed the Surge application on a MICA2 Mote set to a node ID of 2, group ID 125, radio channel 0 (f = 903.018 MHz), and at full power (255).

5.2.7 Alerts

MOTE-VIEW alert manager allows user to define alert condition based on any sensor data of any sensor node. An Alert is a user programmable event that gets triggered when sensor data exceeds pre-defined threshold. An Alert is composed of several pieces:

1. Alert ID – A sequential number to identify the alert
2. Node Name – the node whose conditions can trigger an alert
3. Sensor Name – the sensor whose readings can trigger an alert
4. Alert Condition – the comparison operation to decide whether an alert has triggered
   - Equals
   - Exceeds
   - Below
   - Exceeds or Equals
   - Below or Equals
5. Alert Threshold – the value to compare the alert condition against
6. Unit – the unit for the alert threshold
7. Alert Action – the operation to perform in response to a triggered Alert

There are two possible responses to an Alert:
• Display a message dialog and console message
• Send an email (allows text paging a mobile phone)

8. Alert Interval – specify the interval for the action to be taken.

To set an Alert follow these steps below:

1. Chose from Tools > Alerts > Alert Manager... and click the button “Add New Alert Item”. This operation will add a blank alert item to the list with Alert ID 1 as shown in Figure 5-8.

![Figure 5-8. Screenshot of the Alert Manager.](image)

2. From the Node Name drop-down list, select a node that is currently a part of your deployment.
3. From the Sensor Name drop-down list, select a sensor that you want to use to trigger the Alert. This automatically updates the Unit column associated with that particular sensor type.
4. From Alert Condition drop-down list, specify an appropriate condition to trigger the Alert, viz. “<”, “>”, “=<”, “>=”, “=”.
5. For Alert Threshold field, input numerical value in the text box.
6. From Alert Action drop-down list, chose an action, “Send Alert Email” or “Pop-up Alert Form”.
7. From the Alert Interval drop-down list, specify the interval for the Alert to be sent.
8. From Duration drop-down list, specify the time duration for which the Alert Condition needs to be met before the Alert Action can be triggered.

![Figure 5-9. Screenshot of the Alert Manager after adding two Alerts.](image)
Deleting an Alert is very straightforward, first select and highlight the item that you wish to delete then click the “Delete Alert Item” button, the selected alert item will be deleted.

For the Alert manager to send Email, the users should first configure the mail settings. This can be accomplished as follows:

1. Click from Tools > Alerts > Alert Mail Configure... and that will bring up the dialog box shown in Figure 5-10
2. Enter the SMTP name of your mail server
3. Specify the User Name and Password of your mail account
4. Specify the Email address where the alert needs to be sent
5. Enter any optional message to be sent in the body of the Email
6. Click on OK.

![Figure 5-10. Screenshot of the Alert Mail Configuration.](image)

5.3 Units

The units menu allows the user to select their preferred engineering units for a given class of sensor. All sensors of that class will convert to the units specified by the user. A user may select to view raw data as direct ADC readings instead of a particular unit.

5.3.1 Temperature

_MOTE-VIEW_ supports temperature conversion to Celsius (C), Fahrenheit (F), and Kelvin (K).

5.3.2 Pressure

_MOTE-VIEW_ supports pressure conversion to Atmosphere (atm), Bar (mbar), Pascal (Pa), Per mm of Hg (torr), and Pounds per square inch (psi).
5.3.3 Acceleration

MOTE-VIEW supports acceleration conversion to Meters per second squared (m/sec²), and Relative gravity (g).

5.4 Windows

5.4.1 Server Configuration
This menu will bring up the Server Configuration dialog.

5.4.2 Data
This menu will bring up Data View tab to the front of the display.

5.4.3 Charts
This menu will bring up Chart View tab to the front of the display.

5.4.4 Topology
This menu will bring up Topology View tab to the front of the display.

5.5 Help

5.5.1 MOTE-VIEW Help
This menu will bring up this document, MOTE-VIEW 1.2 User’s Manual.

5.5.2 Tutorial
This menu will bring up a tutorial document for MOTE-VIEW.

5.5.3 Support
This menu will open a browser window to the Crossbow support webpage if your PC is connected to the internet.

5.5.4 About
This menu will bring the about screen and display the exact MOTE-VIEW version number.

5.6 Toolbar Menu

The toolbar provides quick access to some of the more common features of the MOTE-VIEW application. From left to right, the toolbar provides buttons for database connection, data logging with XServe, MoteConfig, Alerts Manager, MOTE-VIEW settings, export to CSV, print, and print preview, refreshing the view, and displaying live (vs. historical) sensor network data.
5.7 Dialogs

5.7.1 Database Server Configuration

Server configuration is a critical first step to using MOTE-VIEW. The Database Server Configuration dialog can be opened in three ways:

a. Click the Connection icon on the Tool bar:

b. Select File > Connect...

c. Select Window > Server Configuration...

From within the Database Server Configuration dialog window, you can enter the following fields:

- **Server**: Can be a hostname or IP address. To specify your workstation as a local server enter `localhost`.

- **Port**: TCP/IP port of PostgreSQL database server. Defaults to 5432 which is the standard PostgreSQL port.
• **User:** username for database server. *tele* is the default.

• **Password:** password for database server. *tiny* is the default.

After typing in the four fields above, click the **Connect** button to attempt to connect to the said database and update the pull down lists for what is available on that server.

To complete server selection, the pull down lists should then be used to specify which result set on the server is to be viewed:

• **Database:** All available *PostgreSQL* databases on the server will be displayed

• **Table Name:** The table files containing results to be viewed. This is populated with available tables.

After selecting a database, table, and client, the user may click **Apply** to have *MOTE-VIEW* display the results data for the selections. **Cancel** will close the dialog window.

Saved configurations are stored in the tree view to the left in a nested fashion. The server names are listed with the database names enclosed within. Clicking a server name opens the list of saved databases on that server. Clicking on a database name loads those settings into the fields on the right. To connect to a saved configuration, click on the database name and click **Apply**.

When a database has multiple result tables, the last saved table will be used.

### 5.7.2 Mote Properties

This dialog is opened by selecting the **Properties** menu when right clicking a node in the Topology map, or by double-clicking a node in the node list. It allows the user to assign a **Name** and **Group** of the Mote.

The **Color** tab allows the user to select the charting color for the Mote.

---

**NOTE:** The **Sensorboard** tab of this dialog is only used to display calibration coefficients for boards that are equipped with an Intersema pressure temperature sensor such as the MTS400, MTS420, and MEP410.
This dialog is opened by selecting the Properties menu when right clicking a node in the Topology map, or by double-clicking a node in the node list.

5.8 Server Messages
The server messages pane is the bottom section of the MOTE-VIEW window and displays server side messages, database errors, and general status messages as MOTE-VIEW is running.
6 Server Administration

XServe can be used as a data logging server and comes preinstalled with MOTE-VIEW. A user may run XServe from a Cygwin command line (included in the TinyOS 1.1.0 installer), or directly within MOTE-VIEW using the data logging dialog accessible from the Tools > Log Data menu.

**NOTE:** To use the command line interface for XServe you must have the Cygwin shell installed on your PC. Instructions for installing Cygwin from TinyOS InstallShield Wizard are detailed in the Getting Started Guide.

6.1 General configuration

Generally, the installation wizard will install XServe automatically. The cygwin1.dll library in the MOTE-VIEW installation directory must exactly match the cygwin1.dll library in the user’s cygwin/bin/ directory for proper operation.

6.2 XServe Logging

XServe is a general purpose command line tool for displaying sensor readings being received by a gateway Mote. If using a Cygwin, type `xserve` –? to display the command line options.

```
$ xserve -?
XServe Ver:$Id: xserve.c,v 1.21 2005/09/02 08:15:37 pipeng Exp $
Using params: [help]
Usage: xserve <-?|r|p|x|l|d|v|q> <-l=table> <-s=device> <-b=baud> <-i=server:port>
  -?  = display help [help]
  -r  = raw display of tos packets [raw]
  -a  = ascii display of tos packets [ascii]
  -p  = parse packet into raw sensor readings [parsed]
  -c  = convert data to engineering units [cooked]
  -l  = log data to database or file [logged]
  -x  = export raw readings in csv spreadsheet format [export]
  -xc = export cooked in csv spreadsheet format [export]
  -d  = debug serial port by dumping bytes [debug]
  -b  = set the baudrate [baud=#|mica2|mica2dot]
  -s  = set serial port device [device=com1]
  -i  = use socket input [inet=host:port]
  -o  = output port for serial forwarder [onet=port]
  -sf = use serial forwarder input [inet=host:port]
  -f  = specify framing (0=auto|1=on|2=off)
  -w  = specify header width [header=offset]
  -h  = html reporting of health [html]
  -t  = display time packet was received [timed]
  -q  = quiet mode (suppress headers)
  -v  = show version of all modules
```
XServe will be installed automatically by the MoteViewSetup installation procedure. Data logging to a database was added to XServe in version 1.11. To start XServe data logging, from a Cygwin command line type in a command similar to the following. Be sure to specify the proper input device (-i) and port.

<table>
<thead>
<tr>
<th>XServe Command Line for Data Logging</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xserve -l</td>
<td>Log readings from a MICA2 as gateway on COM1 (default)</td>
</tr>
<tr>
<td>xserve -l -s=com4</td>
<td>Log readings from a MICA2 as gateway on COM4</td>
</tr>
<tr>
<td>xserve -l -b=mica2dot</td>
<td>Log readings from a MICA2DOT as gateway on COM1 (default)</td>
</tr>
<tr>
<td>xserve -l -r -c -i</td>
<td>Log data to database, display raw packets, and display converted readings from SerialForwarder on localhost port 9001</td>
</tr>
</tbody>
</table>

**NOTE:** Every time you reboot your computer and want to keep logging data, you must open a Cygwin window and type `xserve -l` to start logging data.
7 Database Administration

7.1 PostgreSQL

**NOTE:** To use the command line interface for PostgreSQL you **must** have the Cygwin shell installed on your PC. Instructions for installing Cygwin from TinyOS Install Shield Wizard are detailed in the Getting Started Guide.

PostgreSQL is an advanced relational database system that is provided with the Cygwin on the PC and is available on the Stargate. The database tables that MOTE-VIEW accesses can be manipulated directly by advanced users. To access the PostgreSQL database, from a Cygwin shell type

```bash
psql -h localhost -U tele task
```

Below is an example of what you should get as a response to that command:

```bash
$ psql -h localhost -U tele task
Welcome to psql 7.4.5, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

```

7.2 SQL

SQL is the generic command language used to manipulate databases such as PostgreSQL. SQL commands can be typed in directly from the PostgreSQL command shell. A list of common and useful commands follows:

7.2.1 Display all readings

Type

```sql
select * from <tablename>;
```

The `select` statement will display results out `from` the given `<tablename>`. The `*` character is a wildcard meaning that all columns should be displayed.
7.2.2 Display subset of readings

Type

\[
\text{select field}_1, \text{ field}_2, \ldots \text{ from } <\text{tablename}>
\]
\[
\text{where field}_n \text{ condition value ;}
\]

The \textit{select} statement will display results out \textit{from} the given \textit{<tablename>}. The \texttt{*} character is a wildcard meaning that all columns should be displayed.

7.2.3 Rename a table

Type \texttt{ALTER TABLE <tablename> RENAME TO <newname>;};

7.2.4 Delete all readings from table

Type \texttt{DELETE FROM <tablename>;};

7.2.5 Deleting specific readings from table

To delete all results before the specified date, type

\[
\text{DELETE FROM <tablename> WHERE result_time < '2004-11-20'};
\]

To delete all results with ADC voltage reading greater than 400.

\[
\text{DELETE FROM <tablename> WHERE voltage > 400;};
\]

To delete all results from node number 3.

\[
\text{DELETE FROM <tablename> WHERE nodeid = 3;};
\]

7.2.6 Delete table entirely

Type \texttt{DROP TABLE <tablename>;};
7.3 Database Tools

*PostgreSQL* comes with other tools for offline manipulation of data besides the *psql* shell. The more useful of these are described here. These windows command prompt version of these tools are installed to `C:\Program Files\PostgreSQL\8.0.0-rc1\bin` by default. The *psql* tool is available from the *Cygwin* command prompt as well.

### 7.3.1 PostgreSQL Export

To output entire task database to a file, e.g., `my_database.out`:

```
pg_dump -h localhost -U tele -f my_database.out task
```

To save contents of `surge_results` table to a file of SQL commands named `surge.out`:

```
pg_dump -h localhost -U tele -t surge_results -f surge.out task
```

### 7.3.2 PostgreSQL Import

To load files from a PostgreSQL exported table, use the following command:

```
psql task < surge.out
```